Pulp - Story #6736

Story # 6134 (CLOSED - CURRENTRELEASE): [EPIC] Pulp import/export

As a user, I can export into a series of files of a particular size

05/14/2020 04:43 PM - daviddavis

Status:	CLOSED - CURRENTRELEASE	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	ggainey	**% Done:**	100%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:	3.4.0		
Platform Release:		**Tags:**	
Groomed:	No	**Sprint:**	Sprint 73
Sprint Candidate:	No	**Quarter:**	

Description

This is to support the use case where Katello wants to export a series of files of a given size that can be recombined and imported later.

This would allow Katello to avoid the 3x disk usage problem whereby three copies of an artifact exist: the original one, the exported one, and the one in the split archives.

Related issues:

Related to Pulp - Story #6737: As a user, I can import a split export	**CLOSED - CURRENTRELEASE**

Associated revisions

Revision 44bb0623 - 05/26/2020 10:18 PM - ggainey

Teach exporter to understand, validate, and respect chunk_size= parameter.

Since we can now have multiple-output-files, replaced filename/sha256 columns with output_file_info, a JSONField which is a dictionary of filename: hash pairs. closes #6736

History

#1 - 05/14/2020 04:50 PM - ggainey

IRC conversation that spurred this issue:

```
<partha> jsherrill: jturel ggainey so one of the stories I was looking at regarding import/export was the ability to break stuff into smaller batch sizes . Can ya repository content span more than a dv iso ?
<partha> in size
<partha> I guess a content view verison can can right ?
<ggainey> pulp-export drops a single .tar.gz - prev discussion said splitting would be handled by the caller
<ggainey> a single repo-version can certainly be bigger than 4Gb - consider "export all of RHEL7", for example
<jsherrill> ggainey: i thought we talked about pulp doing that because that was the only way to avoid 3x the disc space
<jsherrill> i know originally we said pulp wouldn't, but then i think that dawned on us
<ggainey> jsherrill: pulp is avoiding the 3x disk space problem, by writing directly to a tar.gz instead of to a filesystem-tree and then tar.gz-ing at the end
<jsherrill> ggainey: right but the entire solution needs to avoid the 3x disk space issue
<ggainey> jsherrill: how can it? katello has its own data to export, on top of/in addition to what pulp owns
<jsherrill> ggainey: we just have a simple metadata file, that could easily ship beside the exported data, or appended to a tar file potentially
<ggainey> but you can't "append to the tarfile" until that file exists.
<jsherrill> right, it just depends on how the files are split and if the format supports appending, but if not like i said we can just ship it beside the file(s)
<jsherrill> i swear we talked about all this ;)
<ggainey> ...
<ggainey> we did - and the outcome (that I heard, at least) was "pulp will produce a .tar.gz and it will be the caller's (ie katello's) reposonsibility to handle splitting it up however it wants
<ggainey> "
<ggainey> which is what I've been saying/documenting ever since
<jsherrill> that was, until we realized we couldn't solve the 3x disk space problem that way
<ggainey> which is why, for example, the API docs have no mention of ever accepting a "chunk/split size"
<jsherrill> i heard it more as a 'future feature'
<jsherrill> i know its not there now
```

```
<jsherrill> i didn't have any expectation of it being there now
<ggainey> so, just for transparency's sake, today has been a full day already of "oh but I meant <something else>" about things that I have been documenting and explaining and emailing about for months now
<ggainey> sigh
<jsherrill> i'm not sure how we avoid the 3x disk space issue without pulp doing it?
<jsherrill> which was the point right?
<jsherrill> the user doesn't care if some sub-system avoids the 3x disk space
<jsherrill> if the overall solution doesn't
<ggainey> the "three times" thing today, as it has been explained to me, is "in pulp, in filesystem-export-tree, in resulting .tar.gz"
<jsherrill> i'm sorry if that was explained incorrectly then :(
<ggainey> one can then use, say, the os 'split' command to cut that tar.gtz up into chunks to move to some medium
<jsherrill> but i mean its possible i'm misremembering this conversation
<ggainey> this is why I basically beg people to write things down/add to docs/answer email threads
<ggainey> it's fine
<jsherrill> i think this was during a meeting?
<jsherrill> when we had this realizatipn
<jsherrill> realization
<jsherrill> sorry that it did not come sooner :)
<ggainey> if it's not in the doc, then I am not currently planning to do it
<jsherrill> during docs/email review
<ggainey> so - is this an immediate requirement?
<jsherrill> i think we have to discuss it?
<ggainey> well, I think the discussion would be about time, resources, and priority - but it has to come *after* we have a definition of the actual use-case
<ggainey> esp since that's where the confucion is happening, not on the scheduling end
<ggainey> confusion, even
<daviddavis> ggainey jsherrill I know we talked about it. it was my understanding that it was a future feature and not something we had to solve right away. who can we ask about whether it needs to be soon?
<jsherrill> daviddavis: and that may have been
<jsherrill> we didn't do a good job of establishing what 'future feature' meant
<jsherrill> which future ;)
<daviddavis> agreed
<partha> I mean if we have a 20GB cv dump and have to split it into 5 on disk ourselves wouldn;t we need 2x the space ?
<jsherrill> 3x includes the original content
<jsherrill> iirc
<partha> good point
<daviddavis> is it not possible to split a file in place?
<ggainey> daviddavis: split, when used against a file, needs the space for the original and the parts, until it's done (iirc)
<jsherrill> it looks like it might be possible with dd
<jsherrill> https://superuser.com/questions/177823/are-there-any-tools-in-linux-for-splitting-a-file-in-place
<jsherrill> its really ugly though ;)
<partha> was looking at the smae lol
<ggainey> aanyway - we can do this, I think, for pulpexport by opening the tarfile 'w|', and streaming it to the split() cmd directly
<jsherrill> its 2x+size of one chunk
<jsherrill> which is a lot better than 3x
<jsherrill> ggainey: would that avoid 3x?
<ggainey> jsherrill: yes, because the tarfile never ends up on disk at all
<ggainey> of course, you're now keeping one 'chunk' in *memory* at a time...
<daviddavis> I filed https://pulp.plan.io/issues/6736
<ggainey> (assuming this works, it prob does but would have to see it working)
<jsherrill> lets keep the conversation going, its possible we can do it on our side given that option with dd
<daviddavis> is that less effort though?
<ggainey> jsherrill: not and prefvent 3x - you don't have the file until export is done
<ggainey> oh wait, I see
<ggainey> you have 2x (in pulp, in file) and then you end up with 2x-plus-chunk
<ggainey> I get it
<ggainey> sorry, slow today
```

#2 - 05/14/2020 04:53 PM - pulpbot

- Status changed from NEW to POST

#3 - 05/14/2020 04:53 PM - daviddavis

- Related to Story #6737: As a user, I can import a split export added

#4 - 05/14/2020 04:56 PM - daviddavis

- Status changed from POST to NEW

#5 - 05/14/2020 07:30 PM - ggainey

Take as an allowed-param export-size=, we can look at using subprocess.run([split, -b, , -a, 4, -u, -,], input=(tar-gz-stdout)) as an approach. More detail to come.

#6 - 05/14/2020 08:34 PM - ggainey

- Status changed from NEW to ASSIGNED

#7 - 05/14/2020 08:36 PM - ggainey

- Assignee set to ggainey

#8 - 05/22/2020 02:29 PM - ggainey

- Sprint set to Sprint 73

#9 - 05/23/2020 01:50 AM - pulpbot

- Status changed from ASSIGNED to POST

PR: https://github.com/pulp/pulpcore/pull/719

#10 - 05/26/2020 10:19 PM - ggainey

- Status changed from POST to MODIFIED

- % Done changed from 0 to 100

Applied in changeset pulpcore|44bb0623a01b7578e2ec442844c7f5849754b237.

#11 - 05/27/2020 09:28 PM - dkliban@redhat.com

- Sprint/Milestone set to 3.4.0

#12 - 05/28/2020 04:20 PM - dkliban@redhat.com

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE