Pulp - Story #4488

As a user, I can upload chunks in parallel

02/28/2019 11:01 AM - daviddavis

Status:	MODIFIED	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	daviddavis	**% Done:**	100%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:	3.0		
Platform Release:		**QA Contact:**	
Blocks Release:		**Complexity:**	
Backwards Incompatible:	No	**Smash Test:**	
Groomed:	Yes	**Verified:**	No
Sprint Candidate:	Yes	**Verification Required:**	No
Tags:	Katello-P1	**Sprint:**	Sprint 55

Description

We're currently using drf-chunked-uploads[0] but it seems like the library has become unmaintained[1] since we adopted. It has some other quirks and missing features too. So I think we should move off of it and roll our code as part of this story.

Solution

Add a design which supports sha256 and parallel uploads of chunks.

Models

Upload

id = UUID
file = File
size = BigIntegerField
user = FK
created_at = DateTimeField
completed_at = DateTimeField

UploadChunk

id = UUID
upload = FK
offset = BigIntegerField
size = BigIntegerField

Workflow

```
# create the upload session
http POST :24817/pulp/api/v3/uploads/ size=10485759 # returns a UUID (e.g. 345b7d58-f1f8-45d9-d354
-82a31eb879bf)
export UPLOAD='/pulp/api/v3/uploads345b7d58-f1f8-45d9-d354-82a31eb879bf/'

# note the order doesn't matter here
http --form PUT :24817$UPLOAD file@./chunkab 'Content-Range:bytes 6291456-10485759/32095676'
http --form PUT :24817$UPLOAD file@./chunkaa 'Content-Range:bytes 0-6291455/32095676'

# view the upload and its chunks
http :24817${UPLOAD}

# complete the upload
http PUT :24817${UPLOAD}commit sha256=037a47d93670e64f2b1038e6f90e4cfd
```

```
# create the artifact from the upload
http POST :24817/pulp/api/v3/artifacts/ upload=$UPLOAD
```

Additional references

https://github.com/douglasmiranda/django-fine-uploader
https://medium.com/box-developer-blog/introducing-the-chunked-upload-api-f82c820ccfcb

[0] https://github.com/jkeifer/drf-chunked-upload
[1] https://github.com/jkeifer/drf-chunked-upload/pull/8

Related issues:

Related to Pulp - Story #4196: As a user, I can upload files in chunks.	**MODIFIED**
Related to Pulp - Test #5263: Test - As a user, I can upload chunks in parallel	**NEW**
Blocks Pulp - Story #4988: As a user, I can remove uploads	**MODIFIED**

Associated revisions

Revision 24b50710 - 06/26/2019 05:53 PM - daviddavis

Add support for parallel chunks and sha256

Also removed drf-chunked-upload.

fixes #4488,#4486

History

#1 - 02/28/2019 11:01 AM - daviddavis

- Related to Story #4196: As a user, I can upload files in chunks. added

#2 - 04/26/2019 10:31 PM - bmbouter

- Tags deleted (Pulp 3)

#3 - 06/17/2019 05:31 PM - daviddavis

- Description updated

#4 - 06/17/2019 06:19 PM - bmbouter

I really like this API. It's legit. I had a few questions I wanted to ask.

What if we didn't have the 'create the upload session' at all? Couldn't the client generate a uuid and start using it?

How do chunks that were never part of an artifact removed?

Should we send a digest value for each chunk? If you have a large file, e.g. many gigs, one incorrect chunk would cause you to upload everything again.

#5 - 06/17/2019 06:31 PM - daviddavis

> What if we didn't have the 'create the upload session' at all? Couldn't the client generate a uuid and start using it?

I see a number of downsides to doing this. First, it's less RESTful. Second, we need to have the total file size before the upload to create the initial file. So we'd have to either pass in the TOTAL file size with the first request (may be hard with parallel uploads) or with every request (kind of awkward).

> How do chunks that were never part of an artifact removed?

I am not totally sure what you're asking but if it's how to remove incomplete uploads, in drf-chunked-uploads they support this (see https://github.com/jkeifer/drf-chunked-upload#settings) but we have yet to leverage this feature. This problem exists currently though and is not needed for this story.

> Should we send a digest value for each chunk? If you have a large file, e.g. many gigs, one incorrect chunk would cause you to upload everything again.

We could definitely add this but I think that's outside the scope of this story. Maybe file another story?

#6 - 06/17/2019 06:33 PM - dkliban@redhat.com

The user should have to start a session so Pulp can have an opportunity to allocate space for the entire upload. Each uploaded chunk can then be written to it's specific place in the file created session creation. This avoids having to write out the whole file when the upload is complete.

Accepting checksums with each uploaded chunk would be helpful.

#7 - 06/17/2019 06:50 PM - bmbouter

We can keep the session creation, it does allow you to make the large file and write into it. If you want to not have chunk checksums initially that is ok too. Also we can worry about the cleanup of uploads that never became Artifacts later too.

#8 - 06/17/2019 07:04 PM - daviddavis

Cool, I filed https://pulp.plan.io/issues/4981 and https://pulp.plan.io/issues/4982.

#9 - 06/18/2019 06:01 PM - daviddavis

- Blocks Story #4988: As a user, I can remove uploads added

#10 - 06/18/2019 06:14 PM - ttereshc

- Groomed changed from No to Yes

- Sprint Candidate changed from No to Yes

#11 - 06/18/2019 06:19 PM - daviddavis

- Status changed from NEW to ASSIGNED

- Assignee set to daviddavis

- Sprint set to Sprint 54

- Tags Katello-P1 added

The changes to the API are blocking Katello who is trying to integrate chunked uploads. Setting P1 tag and adding to sprint.

#12 - 06/20/2019 09:53 PM - daviddavis

- Description updated

#13 - 06/20/2019 10:32 PM - daviddavis

- Description updated

#14 - 06/21/2019 03:26 PM - ttereshc

- Sprint changed from Sprint 54 to Sprint 55

#15 - 06/21/2019 09:13 PM - daviddavis

- Status changed from ASSIGNED to POST

https://github.com/pulp/pulpcore/pull/182

#16 - 06/26/2019 06:23 PM - daviddavis

- Status changed from POST to MODIFIED

- % Done changed from 0 to 100

Applied in changeset pulpcore|24b50710201a5fea73898f8a1fcff286f9e47809.

#17 - 08/13/2019 08:09 PM - kersom

- Related to Test #5263: Test - As a user, I can upload chunks in parallel added