Pulp - Story #2261

As a user, I can see the total size in bytes that a repository's files use on disk

09/15/2016 05:08 PM - mhrivnak

Status:	NEW	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:			
Platform Release:		**QA Contact:**	
Blocks Release:		**Complexity:**	
Backwards Incompatible:	No	**Smash Test:**	
Groomed:	No	**Verified:**	No
Sprint Candidate:	No	**Verification Required:**	No
Tags:	Pulp 2	**Sprint:**	

Description

This should simply be the sum of the size of all files associated with units that are associated with the repository.

This does not include data stored in the database.

For on-demand content, files that are known in the database but have not yet been downloaded should not be counted in the total.

This does not account for the fact that the same unit can appear in multiple repos without incurring additional disk storage use. It will be up to the user to interpret these numbers for individual repos, and consider totals across multiple repos in the context that content may be shared.

A natural way to represent this would be as an attribute of a repository, but that doesn't have to be the implementation if a better option presents itself.

History

#3 - 12/10/2016 06:07 PM - bmbouter

- Checklist item [ ] a release note on this feature added
Checklist item [ ] update any API result examples in the documentation to include the this new attribute in the response added

- Groomed changed from No to Yes

- Sprint Candidate changed from No to Yes

#4 - 12/10/2016 06:16 PM - bmbouter

- Groomed changed from Yes to No

Before grooming can a full example be given of a repo detail view that also shows the new field's being added?

Also what about having it sum them by type and also give a total. For example:

```
"size": {
   "total": 238260,
   "rpm": 227341,
   "drpm: 10919,
 },
```

#5 - 12/13/2016 05:58 PM - mhrivnak

How would we obtain the total size in a generic way? Maybe we could add a "size" attribute to FileContentUnit, and leave it up to plugins to populate that if/when possible. Would this be optional?

What about on_demand? If a file hasn't been downloaded, should the unit's size be 0? It seems more intuitive that it should be the expected size after download. But what about the size of the repo? I'm not sure why, but my intuition is the opposite: that a repo's size should be 0 if none of its units have been downloaded. Maybe the algorithm would roughly be this:

```
size = 0
for unit in repo:
  if unit.downloaded is True:
    size += unit.size
```

For that reason, maybe the repo's attribute would be better-named "disk_use", or something like that.

Presumably the repo's size would have to be updated under any of these circumstances:

- content added (sync/copy/upload)
- content removed (sync/remove)
- unit marked as downloaded

What about "shared content", used by ostree, where multiple units can reference the same files? I wonder if the ostree tooling itself has a standard, "expected" way to represent size of a repo where there may be overlap between branches.

Given the above questions, I wonder if this is complex enough that it should wait for pulp 3.

#6 - 12/13/2016 07:02 PM - bmbouter

+1 to making the attribute name 'disk_use' and by that name I only expect it to count the on-disk of already downloaded units (not on_demand units).

+1 to putting it as an attribute on FileContentUnit.

What if we make the pre_save_handler of FileContentUnit calculate the attribute if it is not already set and the file is downloaded locally. It could default to null otherwise to distinguish against empty files. We do this already for important things so doing it for this would work I think [0].

We would also add a property to Repository that contains the algorithm you posted and that field would be summed at runtime and not formally saved on the Repository. Is that what others are thinking?

Also for shared content units I think having a unit be counted against many repositories that share that unit I think is OK. I think of this feature as helping to answer the question: "If I export or download all units for a repo outside of Pulp how much space do I need?"

Also using the FileContentUnit attribute we could have Pulp sum the total space and available space as part of the /status/ API but that is a separate feature answering a different question: "how much space is Pulp using, and how much does it have available before filling up the filesystem".

+0 to waiting for Pulp3 is file. Regardless, I wanted to express my ideas here anyway which could also be translated directly to Pulp3.

[0]: https://github.com/pulp/pulp/blob/91a1e28c9e7d3dee418d5c7680dbf25c3e7adc63/server/pulp/server/db/model/__init__.py#L849-L851

#7 - 09/14/2018 03:57 PM - amacdona@redhat.com

- Sprint Candidate changed from Yes to No

#8 - 04/15/2019 10:24 PM - bmbouter

- Tags Pulp 2 added