

openQA Project - action #76984

coordination # 64746 (Blocked): [saga][epic] Scale up: Handle large storage efficiently to be able to run current tests efficiently but keep big archives of old results

[epic] Automatically remove assets+results based on available free space

2020-11-04 16:10 - okurz

Status:	Blocked	Start date:	2021-01-21
Priority:	Normal	Due date:	2021-03-09
Assignee:	okurz	% Done:	0%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:	Ready		
Difficulty:			
Description			
Motivation			
<p>See examples like #76822 : openQA has automatic removal of assets+results but the sum of all configured retention periods and asset quotas can still exceed the available space so that manual administration is required. In case the cleanup based on these parameters can not free enough space we should do the next step and remove more until we have enough free space again. We already do something similar in https://gitlab.suse.de/openqa/salt-states-openqa/-/blob/master/etc/master/cron.d/SLES.CRON#L18 to remove videos of older test jobs which we identified as a big contributor to space usage.</p>			
Acceptance criteria			
<ul style="list-style-type: none">• AC1: the filesystem including the openQA results directory is ensured to have at least a configured amount of free space			
Suggestions			
<ul style="list-style-type: none">• Read and understand https://gitlab.suse.de/openqa/salt-states-openqa/-/blob/master/etc/master/cron.d/SLES.CRON#L18• Extend the existing asset+result cleanup to<ul style="list-style-type: none">◦ check the free space of the filesystem including the assets/results directory◦ compare the free space against a configured value, e.g. in openqa.ini◦ if free space is below limit after results cleanup remove more data from results checking in each step until free space limit is reached, e.g.<ul style="list-style-type: none">◦ videos from oldest, non-important jobs first ("oldest first" can mean simply job id numbers ascending order)◦ other results from oldest, non-important jobs◦ videos from oldest, important jobs◦ other results from oldest, important jobs◦ if after all steps free space limit could still not be reached, i.e. if all result data was removed, raise error◦ the above can be configured as well, e.g. "results_free_space_cleanup_components=non-important-results-videos,non-important-results-other,important-results-videos,important-results-other"• can use https://software.opensuse.org/package/perl-Filesys-Df?search_term=perl-FileSys-Df• can mock "df" in tests to simply give back what we want, e.g. "enough free space available" or "free space exceeded"• Optional: Extend to assets as well			
Impact			
<p>This can also greatly help us as administrators of osd to ensure that /results limits are not exceeded which repeatedly caused us additional administration work.</p>			
Workaround			
<p>Have a periodic job calling "df" and checking against limit, remove results otherwise</p>			
Subtasks:			Feedback
action # 88121: Trigger cleanup of results (or assets) if not enough free space based o...			
Related issues:			
Related to openQA Project - action #64881: Reconsider triggering cleanup jobs		Blocked	2020-03-26

Related to openQA Infrastructure - action #68053: powerqaworker-qam-1 fails t...	Workable	2020-06-14	
Copied from openQA Infrastructure - action #76822: Fix /results over-usage on...	Resolved	2020-10-30	2020-11-13

History

#1 - 2020-11-04 16:10 - okurz

- Copied from action #76822: Fix /results over-usage on osd (was: sudden increase in job group results for SLE 15 SP2 Incidents) added

#2 - 2020-11-04 16:11 - okurz

- Description updated

- Status changed from New to Workable

#3 - 2020-11-04 20:27 - okurz

- Related to action #64881: Reconsider triggering cleanup jobs added

#4 - 2020-11-04 20:27 - okurz

- Description updated

#5 - 2020-11-27 21:38 - okurz

- Parent task set to #64746

#6 - 2020-12-08 16:18 - mkittler

- Related to action #68053: powerqaworker-qam-1 fails to come up on reboot (repeatedly) added

#7 - 2020-12-09 15:27 - mkittler

- Assignee set to mkittler

#8 - 2020-12-09 16:21 - mkittler

1. Filesys::Df would be very simple to use: <https://metacpan.org/pod/Filesys::Df>

2. Regarding result cleanup

1. We would likely want to run this after the regular cleanup. That's after all limit_screenshots tasks are done. So I'd add an additional Minion task and would enqueue such a job to run after all limit_screenshots jobs via the parents argument of enqueue (<https://metacpan.org/pod/Minion#enqueue1>).
2. After cleaning up a job we need to check whether we are now beyond the configured limit in order to decide whether we need to proceed and cleanup more jobs. The problem is that the screenshots associated with the deleted job are not immediately deleted when deleting the job. Taking care of dangling screenshots is so far only implemented as a separate task (the one mentioned in 2.) which considers all screenshots. It looks like we need a 2nd way to cleanup screenshots which would not try to do it in batches for all screenshots but only considers the screenshots related to a certain job. Not sure how efficient that would be but maybe it would be ok to run that every time after cleaning up a job while exceeding the free space. Making a query for the screenshots exclusively used by a certain job wouldn't be very difficult but it might be expensive to run, especially since we would possibly need to run it quite often.

3. Regarding asset cleanup

1. We actually already have size limits but over-allocate in practice. So I assume this ticket is about combining the possibility of an over-allocation with a cleanup that ensures we do not actually run out of disk space.
2. The previous point still leaves the question which assets should be deleted first. Maybe a 2nd asset cleanup run should be performed after the regular asset cleanup.
 1. It would use a scaled-down version of the configured quotas. With scaled-down I mean the absolute sizes of each group would be reduced to fit some limit but the proportions would be preserved. So the configured quotas would only serve as a weight factor for the 2nd cleanup.
 2. It would stop immediately if the threshold for the disk utilization is no longer exceeded. So assets are not needlessly removed.
 3. I hope that the combination of the previous points 1. and 2. allows that certain groups can still retain their over-allocated assets as long as enough other groups don't actually use their allocated limit.
3. Maybe it makes sense to visualize the scaled-down limits first within openQA's asset statistics.
4. It would also be nice to be able to perform a dry-run (with production data) before introducing changes like this.

#9 - 2020-12-10 04:07 - openqa_review

- Due date set to 2020-12-24

Setting due date based on mean cycle time of SUSE QE Tools

#10 - 2020-12-10 13:28 - mkittler

Here a few queries related to the screenshots-to-job mapping in our database which can help with point 2.2 from my previous comment:

```
number of screenshots per jobs:
```

```
openqa-local=> select job_id, count(distinct screenshot_id) as screenshot_count from screenshot_links where job_id = 1801 group by job_id;
```

number of jobs referencing a screenshot:

```
select count(distinct job_id) as screenshot_usage from screenshot_links where screenshot_id = 242820;
```

exclusive screenshots per job:

```
select distinct screenshot_id from screenshots join screenshot_links on screenshots.id=screenshot_links.screenshot_id where job_id = 1801 and (select count(job_id) as screenshot_usage from screenshot_links where screenshot_id = id and job_id != 1801) = 0;
```

shared screenshots per job:

```
select distinct screenshot_id, (select count(distinct job_id) as screenshot_usage from screenshot_links where screenshot_id = id and job_id != 1801) as spread from screenshots join screenshot_links on screenshots.id=screenshot_links.screenshot_id where job_id = 1801 and (select count(job_id) as screenshot_usage from screenshot_links where screenshot_id = id and job_id != 1801) > 0 order by spread desc;
```

Of course the query exclusive screenshots per job is the one of interest for this ticket. It runs reasonably fast on my local database. However, on OSD it took so long to execute it that I had to abort it as we have tons of jobs and screenshots there. I suppose the query can still be written in a more optimal way but I wouldn't expect a miracle.

By the way, the distincts in these queries are required because the screenshot_links contains a LOT of duplicates. I'm wondering why we don't have a unique constraint for the pair of screenshot_id and job_id. Even in my local database I see the same job-to-screenshot mapping over 200 times. That's certainly something we might want to improve although it is of course out-of-scope for this ticket.

#11 - 2020-12-10 14:02 - mkittler

The query for exclusive screenshots per job can be easily improved. The following query returns in ~31 ms on OSD which is acceptable:

```
select distinct screenshot_id from screenshots join screenshot_links on screenshots.id=screenshot_links.screenshot_id where job_id = 5147889 and not exists(select job_id as screenshot_usage from screenshot_links where screenshot_id = id and job_id != 5147889 limit 1);
```

~~Without the distinct it goes even down to ~18 ms so if we can cope with duplicates later we could consider avoid using it here.~~ The 2nd run was just faster. The explicit limit 1 can also be omitted because PostgreSQL seems to be smart enough.

#12 - 2020-12-10 15:16 - mkittler

- Description updated

#13 - 2020-12-28 15:00 - cdywan

- Due date changed from 2020-12-24 to 2021-01-08

- Status changed from Workable to Feedback

I suppose this is still being researched, hence setting to *Feedback*. Also bumping the *due date* to account for holidays.

#14 - 2021-01-12 13:34 - cdywan

- Due date deleted (2021-01-08)

#15 - 2021-01-21 10:55 - cdywan

The PR <https://github.com/os-autoinst/openQA/pull/3635>

#16 - 2021-01-21 11:38 - okurz

In today's meeting we discussed a couple of things. One of the last points we mentioned what could be done is to just add the df dependency and trigger the cleanup as soon as df reports not enough space without changing the cleanup implementation. In other words: Whenever new jobs are triggered or would be triggered, call df, compare against configured limit, if not enough free space trigger cleanup and not wait for next periodic, e.g. "nightly", cleanup job. Please split that into a subtask and turn this ticket into epic.

Please for now work under the assumption that calling df is cheap and precise enough.

#17 - 2021-01-21 11:50 - okurz

- Subject changed from Automatically remove assets+results based on available free space to [epic] Automatically remove assets+results based on available free space

created subtask [#76984](#)

#18 - 2021-01-21 13:14 - mkittler

More points from the discussion:

- In the end the "df computation" should be exchangeable with a custom script to return the free percentage to cope with more complicated setups and file systems.
- There could be a dry-run which would run only the video deletion steps (which don't rely on calling df after each deleted job). That would be useful for testing.
- The UI should make it clear that the storage durations are not guaranteed.

#19 - 2021-01-29 14:48 - mkittler

The PR <https://github.com/os-autoinst/openQA/pull/3635> has been merged. I had to remove usages of df during the cleanup. That means it would now be actually easy to provide a dry-run. I think it is worth implementing a dry-run feature so we can enable it in production with more confidence that it won't delete too much. So that would be my next step.

#20 - 2021-01-29 16:48 - mkittler

The dry run is still not that easy after all because the screenshot deletion needed to take into account which jobs *would* have been deleted so far. Maybe I could use a database transaction for that.

I've also noticed that there's one bug I need to fix: So far the size of symlinks (or better their targets) is taken into account but that shouldn't be the case here.

#21 - 2021-02-09 13:17 - mkittler

So far the size of symlinks (or better their targets) is taken into account but that shouldn't be the case here.

A fix for that has already been merged: <https://github.com/os-autoinst/openQA/pull/3705>

As already mentioned, the dry-run would be more work to implement as I thought. It looks like I'd needed to introduce quite some dry-run specific code which would defeat the point of having the dry-run in the first place. So I won't create a PR for that after all. Maybe some people in the team like to help testing the feature by enabling `results_min_free_disk_space_percentage` within `[misc_limits] locally`? It would make sense to check whether `df` returns something that makes sense, e.g. check whether the output of `script/openqa eval -V use Filesys::Df; Filesys::Df::df(OpenQA::Utils::resultdir, 1)` makes sense.

#22 - 2021-02-11 19:53 - okurz

mkittler wrote:

It would make sense to check whether `df` returns something that makes sense, e.g. check whether the output of `script/openqa eval -V use Filesys::Df; Filesys::Df::df(OpenQA::Utils::resultdir, 1)` makes sense.

```
okurz@ariel:~> sudo -u geekotest /usr/share/openqa/script/openqa eval -V 'use Filesys::Df; Filesys::Df::df(OpenQA::Utils::resultdir, 1)'
```

```
{
  "bavail" => '2946475061248',
  "bfree" => '2946475061248',
  "blocks" => '5495946461184',
  "favail" => 2029143627,
  "ffree" => 2029143627,
  "files" => 2147483200,
  "fper" => 6,
  "fused" => 118339573,
  "per" => 46,
  "su_bavail" => '2946475061248',
  "su_blocks" => '5495946461184',
  "su_favail" => 2029143627,
  "su_files" => 2147483200,
  "used" => '2549471399936',
  "user_bavail" => '2946475061248',
  "user_blocks" => '5495946461184',
  "user_favail" => 2029143627,
  "user_files" => 2147483200,
  "user_fused" => 118339573,
  "user_used" => '2549471399936'
}
```

```
okurz@ariel:~> df -h
Filesystem      Size  Used Avail Use% Mounted on
...
/dev/vdb1       5.0T  2.4T  2.7T  47% /space
/dev/mapper/vg0-assets 3.0T  1.8T  1.3T  57% /assets
...
okurz@ariel:~> df
```

```
Filesystem      1K-blocks      Used Available Use% Mounted on
...
/dev/vdb1        5367135216 2489716592 2877418624 47% /space
/dev/vdc         104847360   25452500   79394860   25% /var/lib/pgsql
/dev/mapper/vg0-assets 3219652608 1829882016 1389770592 57% /assets
...
okurz@ariel:~> echo $((2877418624*1024))
2946476670976
```

so, ... yes?

#23 - 2021-02-12 11:54 - mkittler

I guess it makes sense. Note that /space/snapshot-changes/opensuse is on the same partition. Not sure what it is used for but it would of course be problematic if it could possibly fill the entire disk space and might need its own cleanup.

#24 - 2021-02-13 21:57 - okurz

mkittler wrote:

I guess it makes sense. Note that /space/snapshot-changes/opensuse is on the same partition. Not sure what it is used for but it would of course be problematic if it could possibly fill the entire disk space and might need its own cleanup.

True but please consider that out-of-scope. You don't need to care about that, i.e. if df reports below configured threshold, delete results, regardless what service filled up the space.

#25 - 2021-02-15 17:26 - cdywan

Does the above confirmation mean this can be considered done?

#26 - 2021-02-15 18:49 - okurz

you mean if the epic can be resolved? No, we are not there yet.

#27 - 2021-02-16 09:31 - cdywan

okurz wrote:

you mean if the epic can be resolved? No, we are not there yet.

The comments and ACs suggest it's done. Maybe a good idea to reflect here what's still missing.

I would suggest to keep discussions about fixes to subtasks if you're not trying to resolve the epic.

#28 - 2021-02-16 11:24 - okurz

cdywan wrote:

okurz wrote:

you mean if the epic can be resolved? No, we are not there yet.

The comments and ACs suggest it's done. Maybe a good idea to reflect here what's still missing.

I would suggest to keep discussions about fixes to subtasks if you're not trying to resolve the epic.

Maybe you trust in our comments too much. But where do you read that we have the ACs covered? As long as there is no proof that we prevent storage fillup by deleting results based on a configured threshold to keep free the epic is not complete. And we where merely discussing implementation ideas and what "df" reports.

#29 - 2021-02-16 12:35 - cdywan

okurz wrote:

cdywan wrote:

okurz wrote:

you mean if the epic can be resolved? No, we are not there yet.

The comments and ACs suggest it's done. Maybe a good idea to reflect here what's still missing.

I would suggest to keep discussions about fixes to subtasks if you're not trying to resolve the epic.

Maybe you trust in our comments too much. But where do you read that we have the ACs covered? As long as there is no proof that we prevent storage fillup by deleting results based on a configured threshold to keep free the epic is not complete. And we where merely discussing implementation ideas and what "df" reports.

Because we have code that "ensures that we have a configured amount of free space" and it reads to me like you're discussing the existing implementation. Hence, what additional steps are we planning here? Do we want a subticket about overriding df? Or defining the clean-up schedule? Or something else?

#30 - 2021-02-16 19:37 - okurz

cdywan wrote:

Because we have code that "ensures that we have a configured amount of free space"

Well, we need a proof. And for that we need that feature enabled on machines

Do we want a subticket about overriding df?

I don't see what that would bring

Or defining the clean-up schedule?

Maybe, don't know what you mean

Or something else?

Well, in the end we want to have that enabled on both osd+o3. That should all be part of the epic.

#31 - 2021-02-17 10:17 - mkittler

- Assignee deleted (mkittler)

The whole epic is not what I've signed up for.

#32 - 2021-02-17 11:14 - okurz

- Status changed from Feedback to Blocked

- Assignee set to okurz