## openQA Project - action #88121

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

coordination # 76984 (New): [epic] Automatically remove assets+results based on available free space

# Trigger cleanup of results (or assets) if not enough free space based on configuration limit

2021-01-21 11:50 - okurz

| | | | |
|---|---|---|---|
| **Status:** | Resolved | **Start date:** | 2021-01-21 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | mkittler | **% Done:** | 0% |
| **Category:** | Feature requests | **Estimated time:** | 0.00 hour |
| **Target version:** | Ready | | |
| **Difficulty:** | | | |

**Description**

# Motivation

See parent epic #76984 . To be able to progress with #76984 we should try to split out smaller simple stories and start with implementing "df" calls in general. This would also allow us to gather experience if calling df is cheap and reliable enough

# Acceptance criteria

- **AC1:** Regular cleanup of results (or assets) is triggered if free space for results (or assets) is below configured limit
- **AC2:** If no free space limit is configured no df check is called and no cleanup is triggered

# Suggestions

- Extend the existing asset+result cleanup to
  - 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
  - trigger the same cleanup that we would trigger from the systemd timer
- 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

### History

**#1 - 2021-01-28 11:36 - mkittler**

*- Assignee set to mkittler*

This feature is actually orthogonal to the related epic. The only common part is the use of df/statfs. As discussed for the other ticket, we should make that part configurable at some point.

---

I suppose I would create a separate script for this which checks df/statfs (sharing existing code introduced by https://github.com/os-autoinst/openQA/pull/3635). This script would enqueue the cleanup tasks depending on the outcome of the check. The script itself would be invoked by a systemd timer (which a higher frequency than the current timer which enqueues the cleanup unconditionally).

**#2 - 2021-01-28 15:05 - okurz**

mkittler wrote:

> This feature is actually orthogonal to the related epic.

Please explain further what you mean by that. And please do not just point to "what I wrote" :) Is this user story not leading into the same use case?

**#3 - 2021-01-28 15:26 - mkittler**

Your wording "split out" makes it sound like you've took a big ticket and split the task up. However, this ticket seems to be a different task. The only thing in common with the task from the epic is that df/statfs() is involved. I have not been talking about any use-cases here so far but simply about what needs to be done to fulfill the ACs of the tickets.

However, I also don't see the use case for this ticket. Just triggering the current cleanup logic when running out of disk space will not necessarily prevent us from running out of disk space so it isn't really helpful in that regard. This feature could lower the frequency the cleanup is triggered. However, there's no practical benefit. If the cleanup it triggered more often than necessary that shouldn't cause any harm.

**#4 - 2021-01-28 15:28 - mkittler**

Another hint: Coming up with further ideas how to use df/statfs() does not help implementing your previous ideas/tickets. It means *additional* work, not splitting up the work which is already planned or in progress.

**#5 - 2021-01-28 16:49 - okurz**

Actually I think that this story could help to cover #64881 . When we trigger cleanup whenever there is need to cleanup then we do not need to trigger the complete cleanup by timer but instead just call the cheaper "df", right? And the task to call "df" could be done by a periodic minion job instead of a systemd timer so that this functionality needs no further system administration by the user and can work in systemd-less variants as well. The second benefit would be by checking more often that the cleanup hysteresis curve would be smaller and hence less problems with "flickering alerts" in grafana, right?

**#6 - 2021-01-29 04:13 - openqa_review**

*- Due date set to 2021-02-12*

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

**#7 - 2021-02-05 15:06 - mkittler**

> by a periodic minion job instead of a systemd timer

We usually use systemd timers to *enqueue* Minion jobs so they run periodically. I don't know whether it is possible within the Minion framework itself to enqueue jobs periodically. (We can not do that from the main web UI service because it uses preforking. Technically we could enqueue Minion jobs periodically from any of the other services which do not use preforking.)

> When we trigger cleanup whenever there is need to cleanup then we do not need to trigger the complete cleanup by timer but instead just call the cheaper "df", right?

This ticket would certainly help with #64881. However, in the description it would help progressing with #76984 which is a whole different story.

> The second benefit would be by checking more often that the cleanup hysteresis curve would be smaller and hence less problems with "flickering alerts" in grafana, right?

Yes, but again that's not really helping with #76984 which is for extending the cleanup algorithm (to solve running out of disk space due to over-allocation) and this ticket is for just triggering it more often (also to solve running out of disk space but due to running the cleanup not frequently enough and in addition to save resources by avoiding unnecessary cleanups). So these tickets address different problems.

**#8 - 2021-02-05 15:09 - mkittler**

*- Assignee deleted (mkittler)*

I can assign the ticket to me again when we cleared up the questions.

**#9 - 2021-02-15 17:27 - cdywan**

*- Due date deleted (2021-02-12)*

FYI: Unsetting the Due date since that only makes sense when someone's actively working on it.

**#10 - 2021-02-22 10:46 - okurz**

Discussed with mkittler: Next steps

- Add "early-abort" in cleanup jobs based on df-output with configurable minimum space to keep free. With this we re-use the existing systemd time based triggering for the space-aware check
- When we have the early-abort we can configure the systemd timer to trigger more often (instance based) to reduce the hysteresis size
- Trigger cleanup also when creating new jobs, see #64881

**#11 - 2021-02-22 15:11 - mkittler**

*- Assignee set to mkittler*

#### #12 - 2021-02-23 04:06 - openqa_review

*- Due date set to 2021-03-09*

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

#### #13 - 2021-02-23 14:21 - mkittler

Draft PR: https://github.com/os-autoinst/openQA/pull/3750

#### #14 - 2021-02-23 14:21 - mkittler

*- Status changed from Workable to In Progress*

#### #15 - 2021-02-25 13:58 - mkittler

*- Status changed from In Progress to Feedback*

- PR has been merged
- SR for OSD: https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/456
- Configured the same on o3 manually

#### #16 - 2021-03-03 17:07 - mkittler

*- Status changed from Feedback to Resolved*

It seems to work in production:

OSD:

```
  "result" => "Skipping, free disk space on '/var/lib/openqa/share/factory' exceeds configured percentage 20 %
 (free percentage: 22.9447968855517 %)",
```

(result cleanup not skipped)

```
martchus@openqa:~> df -h /assets /results
Dateisystem    Größe Benutzt Verf. Verw% Eingehängt auf
/dev/vdc        7,0T    5,5T  1,6T   79% /assets
/dev/vdd        5,5T    4,5T  1,1T   81% /results
```

o3:

```
  "result" => "Skipping, free disk space on '/var/lib/openqa/share/factory' exceeds configured percentage 20 %
 (free percentage: 29.5415074793063 %)",
  "result" => "Skipping, free disk space on '/var/lib/openqa/testresults' exceeds configured percentage 20 % (
free percentage: 45.2762918429145 %)",
```

```
martchus@ariel:~> df -h /assets /var/lib/openqa/testresults
Dateisystem            Größe Benutzt Verf. Verw% Eingehängt auf
/dev/mapper/vg0-assets  3,0T    2,2T  894G   71% /assets
/dev/vdb1               5,0T    2,8T  2,3T   56% /var/lib/openqa
```

#### #17 - 2021-03-18 13:15 - okurz

*- Due date deleted (2021-03-09)*