

openQA Project - action #91347

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 # 80546 (Blocked): [epic] Scale up: Enable to store more results

[spike][timeboxed:18h] Support for archived jobs

2021-04-19 09:30 - okurz

Status:	Resolved	Start date:	2021-04-19
Priority:	Normal	Due date:	
Assignee:	mkittler	% Done:	0%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:	Ready		
Difficulty:			

Description

Motivation

See ideas in parent [#80546](#)

Suggestions

- add "archived" flag, check what would be needed to read job results based on archived flag
- move one test job to a different path, e.g. /archive/.../\$job
- if archived; read resultdir with configured archive path prefix, not normal resultdir prefix
- add minion job to archive job, triggered on what conditions?
- add feedback in UI "Archived job: Loading can take longer than usual"

Open questions to answer

- which jobs to archive? maybe boolean config: If "archive" then move important jobs to archive if they expire, else delete; or same for all
- when/how to trigger the archiving minion job?
- is it performant enough to have one minion job for one time or one minion job for every job or some bunching in the middle?
- How can admins and users control the archiving decisions?
- Is there a need to unarchive (move back to non-archived results)?
- How to monitor and cleanup from the archive?
- How does the current cleanup encounter archived results, delete them? fail?
- Should we consider "archived" == "important" after some grace time when an important job is moved to a potentially slower archive? So at the time when we trigger the cleanup for non-important jobs we also look at the important jobs and make sure that they are archived or queued for archiving (if that's a more costly process)

History

#1 - 2021-04-19 09:30 - okurz

- Tracker changed from coordination to action

#2 - 2021-04-19 15:26 - mkittler

- Assignee set to mkittler

#3 - 2021-04-20 04:15 - openqa_review

- Due date set to 2021-05-04

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

#4 - 2021-04-22 10:26 - mkittler

- Status changed from Workable to In Progress

Draft implementing some of the suggestions: <https://github.com/os-autoinst/openQA/pull/3858>

#5 - 2021-04-26 13:33 - mkittler

- Status changed from In Progress to Feedback

which jobs to archive? maybe boolean config: If "archive" then move important jobs to archive if they expire, else delete; or same for all when/how to trigger the archiving minion job?

Should we consider "archived" == "important" after some grace time when an important job is moved to a potentially slower archive? So at the time when we trigger the cleanup for non-important jobs we also look at the important jobs and make sure that they are archived or queued for archiving (if that's a more costly process)

The draft PR shows now triggering the archiving during cleanup. It would archive logs of important jobs which are only preserved because the job is considered important. This should be what the last question has asked for. This way of introducing the archiving feature has the advantage that we don't need to introduce new retention periods. The disadvantage is of course that only important jobs benefit from the archiving. However, I suppose we can still improve that later so I'd say it is still a good start.

is it performant enough to have one minion job for one time or one minion job for every job or some bunching in the middle?

The cleanup is already one long Minion job itself so I've been splitting the archiving up. I assume archiving can potentially take a while if lots of jobs are considered at the same time. I suppose it all depends on the I/O performance.

Is there a need to unarchive (move back to non-archived results)?

Not covered so far but it wouldn't be hard to implement the reverse and an admin could trigger it by enqueueing a Minion job manually on the command-line.

How to monitor and cleanup from the archive?

How does the current cleanup encounter archived results, delete them? fail?

I suppose we should add file system checks for the monitoring host in the same way we have them for OSD.

Since I've implemented archiving as an intermediate step of the cleanup (see answer to first question) the archive would be cleaned up as part of the usual cleanup when the important job expires completely.

Screenshots haven't been considered at all because they're shared between jobs which would make things complicated. Of course we could still consider them in the future.

#6 - 2021-04-26 19:57 - okurz

- Status changed from Feedback to Resolved

Perfect. I created three new stories [#91785](#), [#91782](#), [#91779](#) so I think we are good here. Thank you, perfect work! :)

#7 - 2021-04-26 19:58 - okurz

- Due date deleted (2021-05-04)