

openQA Project - action #32545

Provide warning when user creates/triggers misconfigured multi-machine clusters

2018-02-28 15:10 - asmorodskyi

Status: Resolved	Start date: 2018-02-28
Priority: Low	Due date:
Assignee: mkittler	% Done: 0%
Category: Feature requests	Estimated time: 0.00 hour
Target version: Ready	
Difficulty:	

Description

User story

As an openQA operator creating or triggering multi-machine clusters I want to be informed about possible misconfigurations

Acceptance criteria

- **AC1:** An operator is informed when creating or triggering multi-machine clusters in case of multi-machine cluster misconfigurations, e.g. over API response

Suggestions

- Try to create a test with misconfiguration and let tests fail, e.g. as described in [#32545#note-20](#)
- Add corresponding handling of the erroneous case, e.g.
- Forward the error over API and/or web UI where applicable

Original problem observation

support server
<https://openqa.suse.de/tests/1510396/file/autoinst-log.txt> :

```
[2018-02-28T11:51:39.0409 CET] [debug] mutex create 'support_server_ready'
```

parallel job
<https://openqa.suse.de/tests/1510394/file/autoinst-log.txt>

mutex lock 'support_server_ready' unavailable, sleeping 5s - for 2 hours

Original title for the ticket: "Catch multi-machine clusters misconfigured"

Related issues:

Related to openQA Project - action #35140: Parallel job don't see mutex of si...	Resolved	2018-04-18
Related to openQA Project - action #36727: job_grab does not cope with parall...	Resolved	2018-06-04
Related to openQA Tests - action #39128: Misconfigured HA Cluster	Resolved	2018-08-03

History

#1 - 2018-02-28 15:20 - okurz

- Category set to Concrete Bugs

#2 - 2018-02-28 15:45 - pcervinka

Were jobs manually restarted? If so, hpc_mrsh_slave should be restarted to re-trigger all related tests.

If is restarted hpc_mrsh_master or hpc_mrsh_supportserver not all tests are triggered and relation between all jobs is lost.

For example hpc_mrsh_master has relation only to hpc_mrsh_slave after manual restart: <https://openqa.suse.de/tests/1510394#settings>

Question is, is it re-triggering issue in openQA itself or wrong test suite definition?

#3 - 2018-02-28 16:05 - asmorodskyi

I retrigger them one more time let's see if issue will reproduce

#4 - 2018-03-01 07:23 - asmorodskyi

retriggered jobs failed in a same way , and they again have broken dependencies mentioned by @pcervinka. probably this is cause of the problem ? will try to do isos post to see if it will behave the same

#5 - 2018-03-01 07:40 - asmorodskyi

after isos post jobs get correct relations , it might that issue is in only in manual retrigger but let's wait until job finish

#6 - 2018-03-01 08:37 - asmorodskyi

- Priority changed from Urgent to Normal

yes problem is related to manual restart, lower down priority

#7 - 2018-03-05 08:50 - asmorodskyi

- Subject changed from Multi-machine job fail to detect that mutex is created to [sporadic] Multi-machine job fail to detect that mutex is created

- Priority changed from Normal to Urgent

looks like sporadic issue , it hit HPC job group again and this time when jobs was started with isos post Increase priority

#8 - 2018-03-05 09:33 - coolo

- Subject changed from [sporadic] Multi-machine job fail to detect that mutex is created to Catch multi-machine clusters misconfigured

- Priority changed from Urgent to Normal

- Target version set to Ready

the problem is that:

- client had parallel_with supportserver
- server had parallel_with supportserver,client

As such the cluster client+supportserver had no relation to server. This is a misconfiguration we need to catch while posting - but it's not urgent.

#10 - 2018-04-18 10:01 - asmorodskyi

- Related to action #35140: Parallel job don't see mutex of sibling added

#11 - 2018-06-05 14:22 - szarate

- Priority changed from Normal to Urgent

I think this will be needed

#12 - 2018-06-05 14:23 - szarate

- Related to action #36727: job_grab does not cope with parallel cycles added

#13 - 2018-07-16 08:56 - coolo

- Assignee set to coolo

#14 - 2018-07-16 08:56 - coolo

- Target version changed from Ready to Current Sprint

#15 - 2018-08-03 09:40 - szarate

- Related to action #39128: Misconfigured HA Cluster added

#16 - 2018-09-12 09:07 - coolo

- Assignee deleted (coolo)

- Priority changed from Urgent to Normal

- Target version changed from Current Sprint to Ready

#17 - 2019-04-26 09:34 - mkittler

the problem is that:

- client had parallel_with supportserver
- server had parallel_with supportserver,client

As such the cluster client+supportserver had no relation to server. This is a misconfiguration we need to catch while posting - but it's not urgent.

Not sure whether I understand this correctly. Server and supportserver are different tests? What is the expected configuration? How would openQA know what the expected configuration is?

#18 - 2019-04-26 20:11 - asmorodskyi

if you will open HPC job group you will see several MM tests grouped by 3 jobs:

- hpc_mrsh_master, hpc_mrsh_slave, hpc_mrsh_supportserver
- hpc_munge_master, hpc_munge_slave, hpc_munge_supportserver

or

- hpc_ganglia_server, hpc_ganglia_client, hpc_ganglia_supportserver

I agree terminology is confusing but we have what we have :) if you still have questions fill free to ping me in IRC

#19 - 2019-04-29 12:41 - mkittler

Ok, so *_server and *_supportserver are different tests and supposed to run in parallel when this cluster is configured and displayed correctly: <https://openqa.suse.de/tests/2837345#dependencies>

But I'm still not sure what kind of misconfiguration we're looking for and how openQA would be able to detect it.

#20 - 2019-04-29 12:58 - coolo

Meanwhile both _master and _slave are PARALLEL_WITH=supportserver. What was misconfigured was _master had PARALLEL_WITH=supportserver and _slave had PARALLEL_WITH=supportserver,_master

This lead to 2 clusters be formed and one of them had no slave and as the _master already had a parent, _slave ran on its own. What is pretty hard to understand is that with PARALLE_WITH you're setting a parent and there can only be one. But it's possible that the new cluster code already elimates this problem. To be checked, but it's still a settings smell.

#21 - 2019-05-10 12:52 - mkittler

The frontend code I once wrote for the dependency graph handles this situation by simply putting all those jobs into one big cluster: <https://github.com/os-autoinst/openQA/blob/master/lib/OpenQA/WebAPI/Controller/Test.pm#L737>

I'd say it *should* be like this (and that's how I implemented the graph):

- If A has PARALLEL_WITH=X and B has PARALLEL_WITH=X,Y that makes the following cluster: A,B,X,Y
- So although A and Y are not explicitly specified to run in parallel A and Y are part of the same cluster.
- This is not a misconfiguration (if one was aiming for the big cluster A,B,X,Y).
- When I understand the issue correctly, that is also the behavior [asmorodskyi](#) was expecting. So if the scheduler *would* behave like this the issue here wouldn't have been created.
- The terms parent and child are completely interchangeable for parallel dependencies. Regardless in which way you express the relation - the resulting cluster should be the same. So it makes no sense (from the user perspective) to say you're 'setting a parent'. That might happen behind the scenes (and I guess that's what you meant) but after all this is just about tying jobs together.

If the scheduler behaves differently I would consider it a bug.

I can check how the scheduler code behaves as it is right now and maybe fix it. Of course the code which populates the job dependencies database table from the job settings might handle this incorrect, too. So that's also a place to check.

#22 - 2019-05-10 15:31 - mkittler

- Status changed from New to In Progress

- Assignee set to mkittler

- Target version changed from Ready to Current Sprint

[coolo](#) Seems like a misread your example because _master is likely the same as master and the underscore is just from a failed formatting attempt. So the first bullet point would be:

- If A has PARALLEL_WITH=X and B has PARALLEL_WITH=X,A that makes the following cluster: A,B,X

However, that shouldn't change the conclusion.

I briefly tested the creation of the job dependencies in the database and it seems to be correct: <https://github.com/os-autoinst/openQA/pull/2067>

The test exploits the route I created for the dependency graph. So at least in the graph it would actually be shown up as one big cluster. I now need to verify what the scheduler would do.

#23 - 2019-05-11 07:07 - coolo

I don't think the scheduler misbehaves either - but the config doesn't make sense and the job creation basically applies a workaround by ignoring half the settings.

#24 - 2019-05-13 08:37 - mkittler

but the config doesn't make sense

Also in general?

the job creation basically applies a workaround by ignoring half the settings

Not sure what workaround you mean. So settings are lost when creating such a cluster?

#25 - 2019-05-17 14:04 - mkittler

- Status changed from In Progress to New

- Assignee deleted (mkittler)

#26 - 2019-09-03 08:56 - coolo

- Target version changed from Current Sprint to Ready

#27 - 2019-09-03 09:01 - okurz

to me it looks like this is blocked by [#41066](#) as in: Probably no one will pick it up until [#41066](#) is done, e.g. my mkittler. So I suggest mkittler picks the ticket, adds the blocker and sets the status to Blocked. Agreed?

#28 - 2019-09-03 09:32 - coolo

I can't see how this is blocking it.

#29 - 2019-09-03 09:34 - okurz

well, it's not a hard "blocked" but rather a soft work schedule serialization :)

#30 - 2019-09-04 12:02 - okurz

- Status changed from New to Blocked

- Assignee set to okurz

#31 - 2019-10-23 20:24 - okurz

- Status changed from Blocked to New

- Assignee deleted (okurz)

I wonder if this is related to [#32605#note-13](#) . It seems an HPC job was scheduled on its own, not within the cluster as it should.

#32 - 2019-10-25 11:55 - mkittler

So I suggest mkittler picks the ticket

Last time I looked at the ticket it seemed that the user just configured the cluster differently from what was required. But that "different" configuration didn't look generally invalid to me so I didn't know how to proceed with the ticket. The question from my last comment is still unanswered, too. So I will not pick the ticket unless it is clear to me what to do.

#33 - 2019-12-10 13:17 - okurz

[coolo](#) can you clarify regarding [#32545#note-24](#) please

#34 - 2019-12-10 13:27 - coolo

- Category changed from Concrete Bugs to Feature requests

I basically would like to see a warning if you create a one node cluster. Or we go with more forgiving parsing. This feature is purely about user support to get this right, not about correctness of the current implementation.

#35 - 2020-10-09 10:55 - okurz

- Subject changed from Catch multi-machine clusters misconfigured to Provide warning when user creates/triggers misconfigured multi-machine clusters

- Description updated

- Assignee set to mkittler

- Priority changed from Normal to Low

[mkittler](#) can you please check if I phrased the description correctly and set to "Workable" if the answer is "yes".

I would also appreciate feedback from asmorodskyi or any other heavy multi-machine tests user.

#36 - 2020-10-19 11:44 - mkittler

I'll check how it actually behaves at this point. In the best case specifying 2 parallel parents is not a big deal and handled as expected (one still gets one big cluster).

#37 - 2020-10-19 13:18 - mkittler

- File screenshot_20201019_150922.png added

- Status changed from New to Resolved

I've just scheduled a cluster like the one mentioned in [#32545#note-20](#) locally:

- parallel-master: PARALLEL_WITH=parallel-supportserver
- parallel-slave: PARALLEL_WITH=parallel-master,parallel-supportserver
- parallel-supportserver: no dependency variable

Scheduling works fine. It is no problem at all that parallel-slave has two parents. So I think [coolo's](#) statement "What is pretty hard to understand is that with PARALLEL_WITH you're setting a parent and there can only be one." is not true (anymore). The dependency graph has no problems displaying it as well.

The actual job assignment works as expected as well. The scheduler logs Need to schedule 3 parallel jobs for job 1761 (with priority 50) so the jobs are not just by coincidence running in parallel. When cancelling the job parallel-slave which has multiple parents the other jobs are stopped as parallel failed as expected. When restarting it all parallel parents are restarted as well and the whole cluster is replicated as expected.

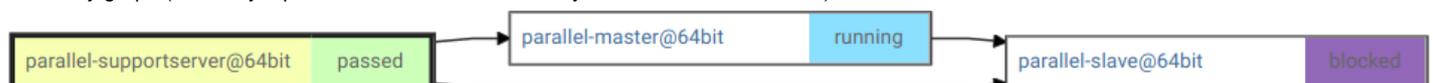
When modifying the cluster so parallel-supportserver has PARALLEL_WITH=parallel-slave things get more interesting because now there's a dependency loop. Theoretically openQA could ignore the loop because the jobs are supposed to run in parallel after all. However, in practice we an error and no jobs are scheduled at all:

```
script/openqa-cli api --host http://localhost:9526 -X POST isos ISO=openSUSE-Tumbleweed-NET-x86_64-Snapshot20201018-Media.iso DISTRI=opensuse ARCH=x86_64 FLAVOR=DVD VERSION=Tumbleweed BUILD=20201018 {"count":0,"failed":[{"error_messages":["There is a cycle in the dependencies of parallel-slave at \hdd\openqa-devel\repos\openQA\script\..\lib\OpenQA\Schema\Result\ScheduledProducts.pm line 698.\n"],"job_id":1767}], "ids": [], "scheduled_product_id":53}
```

I suppose that's ok as well. After all it is quite redundant to specify PARALLEL_WITH on both sides and it is fair to require dropping on of the PARALLEL_WITH variables. The error message is also quite clear and none of the jobs end up running alone (as no jobs are scheduled).

So I would say this issue is not relevant anymore because everything works as expected. There's no need to show warnings about multiple parallel parents when openQA can cope with that.

Of course I was now curious how well it works with other dependencies. So I simply replaced PARALLEL_WITH with START_AFTER_TEST and got this fancy graph (it still says "parallel" because I was too lazy to rename the test suites):



Again, it looks like openQA can cope with multiple parents just fine.

And yes, of course I needed to check what happens with directly chained dependencies as well. Here the cluster does not make much sense because slave can not be started directly after supportserver *and* master. The jobs can nevertheless be scheduled but the cluster is never assigned to a worker leaving master and slave blocked. Judging by the scheduler log, the "cycle detection" for directly chained dependencies catches this case. So I guess we're even good when it comes to directly chained dependencies as nothing unexpected/weird/misleading happens.

It turns out that the cycle between directly chained dependencies causes problems (globally affecting the scheduler) after all. That's of course not within the scope of this ticket.

PR: <https://github.com/os-autoinst/openQA/pull/3490>

#38 - 2020-10-20 21:32 - okurz

awesome assessment, thank you!

Files

screenshot_20201019_150922.png	12.2 KB	2020-10-19	mkittler
--------------------------------	---------	------------	----------