

## openQA Infrastructure - action #30595

### [ppc64le] Deploy bcache on tmpfs workers

2018-01-22 02:29 - yosun

<b>Status:</b> New	<b>Start date:</b> 2018-01-22
<b>Priority:</b> Low	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b> future	
<b>Description</b> qemu-system-ppc64: -monitor telnet:127.0.0.1:20062,server,nowait: Failed to bind socket: Address already in use <a href="https://openqa.suse.de/tests/1405047">https://openqa.suse.de/tests/1405047</a> <a href="https://openqa.suse.de/tests/1405047/file/autoinst-log.txt">https://openqa.suse.de/tests/1405047/file/autoinst-log.txt</a>	
<b>Related issues:</b>	
Related to openQA Project - action #30388: [qam] openqaworker10:4 - worker f...	<b>Resolved</b> 2018-02-26
Related to openQA Infrastructure - action #38780: [tools] Remove tmpfs from m...	<b>Rejected</b> 2018-07-24
Related to openQA Project - coordination #64746: [saga][epic] Scale up: Effic...	<b>Blocked</b> 2018-10-02
Has duplicate openQA Tests - action #30249: [sle][migration][sle15][ppc64le] ...	<b>Resolved</b> 2018-01-12

#### History

##### #1 - 2018-01-22 05:47 - okurz

- Subject changed from [tools] Address competition(Failed to bind socket: Address already in use) to [tools][ppc64le][malbec] Address competition(Failed to bind socket: Address already in use)

- Category set to 168

- Status changed from New to In Progress

- Assignee set to okurz

Could be a duplicate of #30388 . <https://openqa.suse.de/tests/1403769> is the last test that did not incomplete. <https://openqa.suse.de/tests/1403769/file/autoinst-log.txt> shows

```
[2018-01-20T16:57:14.0206 CET] [debug] no [2018-01-20T16:57:14.0207 CET] [debug] waitpid for 37981 returned 0
[2018-01-20T16:57:14.0207 CET] [debug] capture loop failed Can't write to log: No space left on device at /usr
/lib/os-autoinst/bmwqemu.pm line 197.
```

```
DIE Can't write to log: No space left on device at /usr/lib/os-autoinst/bmwqemu.pm line 197.
```

```
at /usr/lib/os-autoinst/backend/baseclass.pm line 80.
  backend::baseclass::die_handler('Can't write to log: No space left on device at /usr/lib/os-a...') called
at /usr/lib/perl5/5.18.2/Carp.pm line 100
  Carp::croak('Can't write to log: No space left on device') called at /usr/lib/perl5/vendor_perl/5.18.2/Mo
jo/Log.pm line 32
```

I guess that is the problem.

Right now there is no space depletion on malbec (anymore?). I terminated the still running qemu process and will restart the incomplete jobs.

##### #2 - 2018-01-22 05:48 - okurz

- Related to action #30388: [qam] openqaworker10:4 - worker fail all tests - qemu instances are left behind added

##### #3 - 2018-01-22 08:38 - okurz

- Has duplicate action #30249: [sle][migration][sle15][ppc64le] test fails in install\_and\_reboot - failed by disk space is exhausted added

##### #4 - 2018-01-22 08:39 - okurz

- Priority changed from Normal to Urgent

##### #5 - 2018-01-22 08:42 - okurz

- Assignee deleted (okurz)

- Priority changed from Urgent to High

About 10 jobs are not incompleted now on [malbec](#) . I wonder if certain scenarios, e.g. migration are more prone to deplete the space on malbec. I did the "urgent" part for now, back to backlog for further consideration. I recommend to either further limit the numbers of instances or increase the available space in `/var/lib/openqa/pool`. Right now it's tmpfs 80G 46G 35G 57% `/var/lib/openqa/pool` so no big margin.

**#6 - 2018-01-22 09:07 - coolo**

- Subject changed from `[tools][ppc64le][malbec] Address competition(Failed to bind socket: Address already in use)` to `[ppc64le] Deploy bcache on tmpfs workers`

- Status changed from In Progress to New

malbec does most of its stuff in tmpfs and is overbooked. So if we have too many jobs running in parallel there, we will run into this. But as we're still short on power workers, we have to live with this and restart the jobs that ended up running in parallel.

Alex suggested using bcache to come over this situation, where bcache would use a file on disk to backstore.

**#7 - 2018-01-22 09:09 - coolo**

- Target version set to Ready

**#8 - 2018-02-07 08:38 - mitiao**

- Assignee set to mitiao

**#9 - 2018-02-09 04:23 - mitiao**

- Status changed from New to In Progress

Reproduced issue locally.

Trying to deploy bcache on a test machine following the documents

**#10 - 2018-02-09 09:39 - mitiao**

- Target version changed from Ready to Current Sprint

**#11 - 2018-02-28 05:30 - mitiao**

Tried some deployments on local x86\_64 machines with different configurations.

Wanted to see the inside of malbec and then deploy bcache according to its hardware and os environment.

Currently blocked by 'dead' malbec and waiting for it up.

Filed a infra ticket (107529) for this issue:

<https://infra.nue.suse.com/Ticket/Display.html?id=107529>

**#12 - 2018-02-28 05:44 - coolo**

Don't hack this into malbec - hack this into the salt recipes.

**#13 - 2018-02-28 08:29 - mitiao**

coolo wrote:

Don't hack this into malbec - hack this into the salt recipes.

Got it, will look into deploy by salt

**#14 - 2018-03-08 08:55 - mitiao**

malbec issue has no any update since 2nd of March:

2018-03-01 10:44 The RT System itself - Status changed from 'new' to 'open'

2018-03-01 10:44 Irupp (Lars Vogdt) - Given to mcaj (Martin Caj)

2018-03-01 10:44 Irupp (Lars Vogdt) - Queue changed from BuildOPS to infra

2018-03-01 14:30 mcaj (Martin Caj) - Given to mmaher (Maximilian Maher)

2018-03-02 11:25 fghariani (Fatma Ghariani) - Stolen from mmaher (Maximilian Maher)

2018-03-02 11:25 fghariani (Fatma Ghariani) - Given to ihno (Ihno Krumreich)

Also setup a salt master and minion on local to compose sls file which is for future bcache deployment

**#15 - 2018-03-14 07:30 - mitiao**

Done a sls file to deploy backing device in a looped file and cache device on a x86\_64 test machine.  
Looked into alive power8 worker to get disk configurations, adapting sls to match the real worker.

#### #16 - 2018-03-15 18:52 - szarate

An update, <https://fsp1-malbec.arch.suse.de/cgi-bin/cgi> is now accessible again but doesn't allow ipmi or any other type of interaction apparently.

However the machine seems to have booted in baremetal mode OR there might be network issues that don't allow the ip to be set... who knows.

#### #17 - 2018-03-19 06:59 - mitiao

I am adapting bcache sls for Power8-4 worker, and some questions were pop-up:

1. I can't find any ssd device on the machine, so where the caching device should create?
2. /var/lib/openqa is mounted on a partition, is it ok to format the partition as backing device and mount to /var/lib/openqa? so it would clean up all existing workers and re-install them on bcache device. or use looped file as backing device for newly added workers?

#### #18 - 2018-03-19 07:09 - coolo

if we had ssd in these workers, we wouldn't run them on tmpfs. And we don't want to cache onto ssds, but use the backing device only for overcommitting the memory.

Don't use the full /var/lib/openqa because we don't want to store /var/lib/openqa/cache in memory I guess. But you can shrink it to gain space, but possibly we even other disks.

#### #19 - 2018-03-22 02:40 - mitiao

wip pr submitted:

[https://gitlab.suse.de/openqa/salt-states-openqa/merge\\_requests/37](https://gitlab.suse.de/openqa/salt-states-openqa/merge_requests/37)

#### #20 - 2018-03-22 09:23 - mitiao

malbec is back, last update of <https://infra.nue.suse.com/Ticket/Display.html?id=107529>:  
2018-03-21 11:47 jloeser (Jan Loeser) - Status changed from 'open' to 'resolved'

#### #21 - 2018-03-27 06:38 - mitiao

Did I misunderstand that we don't need make cache on malbec if no ssd device?

As I talked with Coly, only using backing device is not quite different with not using bcache.

Also i don't get the idea about to reduce memory overcommitting by using bcache, since bcache is mainly to use fast device act as a cache for some slower storage device, and the memory usage would be getting larger along with ssd size increase.

And if a cache is needed and no ssd device, cache on hdd is not meaningful, if cache on tmpfs which is still using memory as cache but also limited by memory size and memory is twice used.

btw, since malbec is alive, further work should be specific on malbec.

#### #22 - 2018-03-27 23:04 - szarate

I'm guessing that the solution that coolo is referring to, and that alex pointed him to, was to use the bcache as backing device that would extend the memory of the system via zram... this would extend the amount of memory available in the pool, so that the actual data that is commonly used is kept in the faster ram while the other is moved to the zram-bcache frankenstein... but my memory is foggy here.

[mitiao](#), the idea with using rotating disks, is that since we have some ram to use, just to make sure that there's enough extra space to store big files. Speed is second to some extent here.

Gitlab PR looks fine so far... but well, let malbec be the play ground unless coolo says otherwise :)

#### #23 - 2018-04-03 10:38 - coolo

This is not about zram, no. We have enough memory - in general. But we need to have a strategy where to put the blocks in case the RAM is overcommitted. If bcache can't be used with tmpfs, then we need to find a different solution to have RAM as backing store to the pools.

#### #24 - 2018-04-10 13:09 - algraf

coolo wrote:

This is not about zram, no. We have enough memory - in general. But we need to have a strategy where to put the blocks in case the RAM is overcommitted. If bcache can't be used with tmpfs, then we need to find a different solution to have RAM as backing store to the pools.

You can just use a genuine ramdisk rather than tmpfs as backing store for bcache. That gives you the same effect as the ZRAM variant, but without RAM overhead reduction. The big benefit of bcache that we found was that it nicely continuously streams dirty data onto the backing storage, so it actually saturates the links rather than generate big latency spikes when flushing from time to time.

**#25 - 2018-04-10 14:12 - szarate**

- File *bcache-zram.sh* added

Perhaps the script might shed some more light

**#26 - 2018-04-23 14:02 - szarate**

It was agreed during the sprint planning to move forward with this.

**#27 - 2018-04-23 14:03 - szarate**

[mitiao](#) are you going to keep working on this?

**#28 - 2018-05-23 09:01 - szarate**

- Status changed from *In Progress* to *New*

- Assignee deleted (*mitiao*)

- Priority changed from *High* to *Normal*

- Target version changed from *Current Sprint* to *Ready*

We need to come up with another idea, so most likely this will come back soon

**#29 - 2018-07-24 12:06 - szarate**

- Related to action *#38780: [tools] Remove tmpfs from malbec again and give it a try* added

**#30 - 2018-11-23 14:38 - mkittler**

- Project changed from *openQA Project* to *openQA Infrastructure*

- Category deleted (*168*)

seems to be an infra issue

**#31 - 2020-03-02 21:30 - okurz**

algraf wrote:

You can just use a genuine ramdisk rather than tmpfs as backing store for bcache. That gives you the same effect as the ZRAM variant, but without RAM overhead reduction.

Do you mean ramfs rather than tmpfs or something like `mke2fs -m 0 /dev/ram0` ? Wouldn't in both cases the system memory be prone to run full and cause more havoc than a size-limited tmpfs?

**#32 - 2020-04-21 07:50 - okurz**

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

**#33 - 2020-08-06 11:34 - okurz**

- Tags changed from *caching, openQA, sporadic, arm, ipmi, worker* to *worker*

**#34 - 2020-10-09 07:32 - okurz**

- Priority changed from *Normal* to *Low*

**#35 - 2020-11-04 16:27 - okurz**

- Target version changed from *Ready* to *future*

## Files

---

bcache-zram.sh	2.18 KB	2018-04-10	szarate
----------------	---------	------------	---------