Arvados - Bug #10700

[Crunch2] crunch-dispatch-slurm pileup

12/09/2016 07:48 PM - Peter Amstutz

Status:	Resolved	**Start date:**	01/27/2017
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2017-02-01 sprint		

Description

Crunch-dispatch-slurm got stuck in a loop with a job it couldn't fulfill. This is not reported to the user, so the user has no way of knowing what the problem is.

As a secondary issue, this condition also results in a resource leak of file descriptors. When this happens, things start to pile up.

Finally, when it gets into this condition, a TERM signal stops processing of containers but it deadlocks and does not shut down gracefully.

```
2016-12-08_22:40:56.21879 2016/12/08 22:40:56 Monitoring container 9tee4-dz642-5s5vdaeeu98qhbr
 started
2016-12-08_22:40:56.28660 2016/12/08 22:40:56 Error submitting container
9tee4-dz642-5s5vdaeeu98qhbr to slurm: Container submission failed: [sbatch --share --w
orkdir=/tmp --job-name=9tee4-dz642-5s5vdaeeu98qhbr --mem-per-cpu=100000 --cpus-per-task=1]: exit s
tatus 1 (stderr: "sbatch: error: Batch job submission failed:
 Requested node configuration is not available\n")
2016-12-08_22:40:56.59802 2016/12/08 22:40:56 About to submit queued container
9tee4-dz642-5s5vdaeeu98qhbr
2016-12-08_22:40:56.59806 2016/12/08 22:40:56 sbatch starting: ["sbatch" "--share" "--workdir=/tmp
" "--job-name=9tee4-dz642-5s5vdaeeu98qhbr" "--mem-per-cpu=100
000" "--cpus-per-task=1"]
2016-12-08_22:40:56.71585 2016/12/08 22:40:56 Monitoring container 9tee4-dz642-5s5vdaeeu98qhbr
 finished
2016-12-08_22:40:56.21879 2016/12/08 22:40:56 Monitoring container 9tee4-dz642-5s5vdaeeu98qhbr
 started
2016-12-08_22:40:56.28660 2016/12/08 22:40:56 Error submitting container
9tee4-dz642-5s5vdaeeu98qhbr to slurm: Container submission failed: [sbatch --share --w
orkdir=/tmp --job-name=9tee4-dz642-5s5vdaeeu98qhbr --mem-per-cpu=100000 --cpus-per-task=1]: exit s
tatus 1 (stderr: "sbatch: error: Batch job submission failed:
 Requested node configuration is not available\n")
2016-12-08_22:40:56.59802 2016/12/08 22:40:56 About to submit queued container
9tee4-dz642-5s5vdaeeu98qhbr
2016-12-08_22:40:56.59806 2016/12/08 22:40:56 sbatch starting: ["sbatch" "--share" "--workdir=/tmp
" "--job-name=9tee4-dz642-5s5vdaeeu98qhbr" "--mem-per-cpu=100
000" "--cpus-per-task=1"]
2016-12-08_22:40:56.71585 2016/12/08 22:40:56 Monitoring container 9tee4-dz642-5s5vdaeeu98qhbr
 finished
2016-12-08_22:40:56.88625 2016/12/08 22:40:56 Monitoring container 9tee4-dz642-5s5vdaeeu98qhbr
 started
2016-12-08_22:40:57.04602 2016/12/08 22:40:57 About to submit queued container
9tee4-dz642-5s5vdaeeu98qhbr
2016-12-08_22:40:57.04605 2016/12/08 22:40:57 sbatch starting: ["sbatch" "--share" "--workdir=/tmp
" "--job-name=9tee4-dz642-5s5vdaeeu98qhbr" "--mem-per-cpu=100
000" "--cpus-per-task=1"]
2016-12-08_22:40:57.09960 2016/12/08 22:40:57 Error submitting container
9tee4-dz642-5s5vdaeeu98qhbr to slurm: Container submission failed: [sbatch --share --w
orkdir=/tmp --job-name=9tee4-dz642-5s5vdaeeu98qhbr --mem-per-cpu=100000 --cpus-per-task=1]: exit s
tatus 1 (stderr: "sbatch: error: Batch job submission failed:
 Requested node configuration is not available\n")
2016-12-08_22:40:57.20721 2016/12/08 22:40:57 Error locking container 9tee4-dz642-5s5vdaeeu98qhbr
: "Post https://9tee4.arvadosapi.com/arvados/v1/containers/9tee4-dz642-5s5vdaeeu98qhbr/lock: dial
tcp 10.100.32.5:443: socket: too many open files"
2016-12-08_22:40:57.20723 2016/12/08 22:40:57 Monitoring container 9tee4-dz642-5s5vdaeeu98qhbr
```

```
 finished
2016-12-08_22:40:57.22284 2016/12/08 22:40:57 Error unlocking container
9tee4-dz642-5s5vdaeeu98qhbr
: "Post https://9tee4.arvadosapi.com/arvados/v1/containers/9tee4-dz642-5s5vdaeeu98qhbr/unlock: dia
l tcp 10.100.32.5:443: socket: too many open files"
2016-12-08_22:40:57.48667 2016/12/08 22:40:57 Error running squeue: fork/exec /usr/bin/squeue: too
 many open files
2016-12-08_22:40:57.63356 2016/12/08 22:40:57 Error locking container 9tee4-dz642-5s5vdaeeu98qhbr
: "Post https://9tee4.arvadosapi.com/arvados/v1/containers/9tee4-dz642-5s5vdaeeu98qhbr/lock: dial
tcp 10.100.32.5:443: socket: too many open files"
2016-12-08_22:40:57.64150 2016/12/08 22:40:57 Error unlocking container
9tee4-dz642-5s5vdaeeu98qhbr
: "Post https://9tee4.arvadosapi.com/arvados/v1/containers/9tee4-dz642-5s5vdaeeu98qhbr/unlock: dia
l tcp 10.100.32.5:443: socket: too many open files"
2016-12-08_22:40:57.92863 2016/12/08 22:40:57 Error unlocking container
9tee4-dz642-5s5vdaeeu98qhbr
: "Post https://9tee4.arvadosapi.com/arvados/v1/containers/9tee4-dz642-5s5vdaeeu98qhbr/unlock: dia
l tcp 10.100.32.5:443: socket: too many open files"
2016-12-08_22:40:57.92868 2016/12/08 22:40:57 Error unlocking container
9tee4-dz642-5s5vdaeeu98qhbr
: Post https://9tee4.arvadosapi.com/arvados/v1/containers/9tee4-dz642-5s5vdaeeu98qhbr/unlock: dial
 tcp 10.100.32.5:443: socket: too many open files
2016-12-08_22:40:57.92870 2016/12/08 22:40:57 Error submitting container
9tee4-dz642-5s5vdaeeu98qhbr
 to slurm: Container submission failed: [sbatch --share --workdir=/tmp --job-name=9tee4-dz642-5s5v
daeeu98qhbr --mem-per-cpu=100000 --cpus-per-task=1]: exit status 1 (stderr: "sbatch: error: Batch
job submission failed: Requested node configuration is not available\n")
2016-12-08_22:40:57.92897 2016/12/08 22:40:57 Error unlocking container
9tee4-dz642-5s5vdaeeu98qhbr
: "Post https://9tee4.arvadosapi.com/arvados/v1/containers/9tee4-dz642-5s5vdaeeu98qhbr/unlock: dia
l tcp 10.100.32.5:443: socket: too many open files"
2016-12-08_22:40:57.98657 2016/12/08 22:40:57 Error creating stderr pipe for squeue: pipe2: too ma
ny open files
2016-12-08_22:40:58.04552 2016/12/08 22:40:58 Error unlocking container
9tee4-dz642-5s5vdaeeu98qhbr
: "arvados API server error: #<ArvadosModel::InvalidStateTransitionError: ArvadosModel::InvalidSta
teTransitionError> (422: 422 Unprocessable Entity) returned by 9tee4.arvadosapi.com"
2016-12-08_22:40:58.04554 2016/12/08 22:40:58 Error unlocking container
9tee4-dz642-5s5vdaeeu98qhbr
: arvados API server error: #<ArvadosModel::InvalidStateTransitionError: ArvadosModel::InvalidStat
eTransitionError> (422: 422 Unprocessable Entity) returned by 9tee4.arvadosapi.com
.
.
.
2016-12-08_22:43:15.19059 2016/12/08 22:43:15 Error unlocking container
9tee4-dz642-5s5vdaeeu98qhbr: "arvados API server error: #<ArvadosModel::InvalidStateTra
nsitionError: ArvadosModel::InvalidStateTransitionError> (422: 422 Unprocessable Entity) returned
by 9tee4.arvadosapi.com"
2016-12-08_22:43:15.87004 2016/12/08 22:43:15 Error unlocking container
9tee4-dz642-5s5vdaeeu98qhbr: "arvados API server error: #<ArvadosModel::InvalidStateTra
nsitionError: ArvadosModel::InvalidStateTransitionError> (422: 422 Unprocessable Entity) returned
by 9tee4.arvadosapi.com"
2016-12-09_17:18:27.86690 2016/12/09 17:18:27 Caught signal: terminated
2016-12-09_19:40:16.28603 Stopping crunch-dispatch-slurm
```

Subtasks:

Task # 10991: Review 10700-dispatch	**Resolved**

Related issues:

Related to Arvados - Bug #10701: [Crunch2] crunch-dispatch-slurm leaks file d...	**Resolved**	**01/31/2017**
Related to Arvados - Bug #10702: [Crunch2] crunch-dispatch-slurm buggy error ...	**Resolved**	**02/01/2017**
Related to Arvados - Bug #10703: [Crunch2] crunch-dispatch-slurm deadlocks in...	**Resolved**	**01/31/2017**
Related to Arvados - Bug #10704: [Crunch2] sbatch submit failures not reporte...	**Resolved**	**01/31/2017**
Related to Arvados - Bug #10705: [Crunch2] [API] return a more specific 422 e...	**New**	**01/11/2017**
Related to Arvados - Bug #10729: [Crunch2] Propagate error messages if sbatch...	**New**	**12/14/2016**

Associated revisions

Revision c3d4f8a5 - 01/31/2017 09:32 PM - Tom Clegg

Merge branch '10700-dispatch'

closes #10700
refs #10701
closes #10702
closes #10703
closes #10704

Conflicts:
services/crunch-dispatch-slurm/crunch-dispatch-slurm.go

Revision 8e7c3b36 - 02/09/2017 08:35 AM - Tom Clegg

Merge branch '10700-dispatch'

refs #10700

History

#1 - 12/09/2016 08:01 PM - Peter Amstutz

- Description updated

#2 - 12/09/2016 08:02 PM - Peter Amstutz

- Description updated

#3 - 12/09/2016 08:15 PM - Peter Amstutz

- File current added

#4 - 12/13/2016 08:47 PM - Tom Morris

- Assigned To set to Tom Morris

- Target version set to Arvados Future Sprints

#5 - 12/14/2016 09:00 PM - Peter Amstutz

- Target version changed from Arvados Future Sprints to 2017-01-04 sprint

#6 - 01/04/2017 08:09 PM - Tom Morris

- Assigned To changed from Tom Morris to Peter Amstutz

#7 - 01/04/2017 08:13 PM - Peter Amstutz

- Target version changed from 2017-01-04 sprint to 2017-01-18 sprint

#8 - 01/18/2017 08:10 PM - Peter Amstutz

- Target version changed from 2017-01-18 sprint to 2017-02-01 sprint

#9 - 01/18/2017 08:29 PM - Tom Clegg

- Assigned To changed from Peter Amstutz to Tom Clegg

#10 - 01/26/2017 07:50 PM - Tom Clegg

10700-dispatch @ e34a5060cfc1cc4821b431e8aa6778a31898e0eb

- 10701 (leaking fds) → 318dd88 10701: Remove unneeded complexity in squeue invocation. **(this is an improvement that might be related, but not convincing as a bug fix)**
- 10702 (buggy error handling) → e356309 10703: Do not catch signals in crunch-dispatch-slurm. Simplify "stop dispatcher loop" API. *(simplifying the dispatcher loop fixes multiple concurrent attempts of the same container and resulting log confusion)
- 10703 (deadlock) → e356309 again (SIGTERM just exits immediately now)
- 10704 (sbatch loops too fast) → ae5eb12 10704: Rate-limit startup attempts per container.
- 10705 (better error message for "Unlock unlocked container") → not addressed yet

Couple of other cleanup things included too as separate commits.

#11 - 01/26/2017 08:16 PM - Tom Clegg

- Status changed from New to In Progress

#12 - 01/27/2017 09:42 PM - Peter Amstutz

Tom Clegg wrote:

> 10700-dispatch @ e34a5060cfc1cc4821b431e8aa6778a31898e0eb
>
> - 10701 (leaking fds) → 318dd88 10701: Remove unneeded complexity in squeue invocation. **(this is an improvement that might be related, but not convincing as a bug fix)**

Much nicer. I have to confess I haven't been writing enough Go code lately and have been missing some of the subtle tricks.

- 10702 (buggy error handling) → e356309 10703: Do not catch signals in crunch-dispatch-slurm. Simplify "stop dispatcher loop" API. *(simplifying the dispatcher loop fixes multiple concurrent attempts of the same container and resulting log confusion)
- 10703 (deadlock) → e356309 again (SIGTERM just exits immediately now)

This seems to be the primary bug. This seems to explain the flip-flopping between monitored / unmonitored and the resulting pile up of redundant squeue invocations which is consistent with my initial analysis.

- 10704 (sbatch loops too fast) → ae5eb12 10704: Rate-limit startup attempts per container.

IIRC the Go "Ticker" sends a heartbeat every N seconds so for N=10 and it takes 9 seconds to do something, you'll only get 1 second of sleep before getting another tick. So squeue could still get called twice in < PollPeriod. Suggest MinRetryPeriod should have the same default as PollPeriod, instead of being disabled by default.

- 10705 (better error message for "Unlock unlocked container") → not addressed yet

> Couple of other cleanup things included too as separate commits.

683f537 → this is just an argument of code style. In the pursuit of reducing indentation, the conditional has been transformed from "do this if X is true" to "don't do this if X is not true" which is much harder to reason about. Maybe at least clarify it with comments?

#13 - 01/30/2017 09:30 PM - Tom Clegg

Peter Amstutz wrote:

> This seems to be the primary bug. This seems to explain the flip-flopping between monitored / unmonitored and the resulting pile up of redundant squeue invocations which is consistent with my initial analysis.

I don't remember seeing anything about redundant invocations. Is this written down somewhere? AFAICT the squeue goroutine can't run more than one squeue at a time.

I see I wrote "squeue" in the commit message about simplifying exec.Cmd usage, but "sbatch" was the thing I actually fixed. So... did you also really mean "sbatch" here?

> - 10704 (sbatch loops too fast) → ae5eb12 10704: Rate-limit startup attempts per container.
>
> IIRC the Go "Ticker" sends a heartbeat every N seconds so for N=10 and it takes 9 seconds to do something, you'll only get 1 second of sleep before getting another tick. So squeue could still get called twice in < PollPeriod. Suggest MinRetryPeriod should have the same default as PollPeriod, instead of being disabled by default.

MinRetryPeriod only affects how soon a container will be resubmitted with sbatch after disappearing from the slurm queue, and doesn't use a ticker.

I don't think it's a problem for squeue to be called twice in < PollPeriod -- are you seeing something I'm not?

I do see squeue.go could use some of the same exec.Cmd simplifying treatment noted above, though. I'll do that. (I don't see an FD leak there either, unfortunately.)

> 683f537 → this is just an argument of code style. In the pursuit of reducing indentation, the conditional has been transformed from "do this if X is true" to "don't do this if X is not true" which is much harder to reason about. Maybe at least clarify it with comments?

Fair point. Backed this out half way -- now it's "if A && (B || C)" instead of two nested ifs.

#14 - 01/31/2017 12:24 AM - Tom Clegg

Refactored the squeue stuff, (I think) it's much simpler now.

I didn't see any leaking FDs in the old code, but I figure it can't hurt to get rid of all the pipes and stuff.

10700-dispatch @ a19ffad966b25b3869e666f749f7c6da187bef68

#15 - 01/31/2017 04:33 PM - Peter Amstutz

Tom Clegg wrote:

> Peter Amstutz wrote:
>
> > This seems to be the primary bug. This seems to explain the flip-flopping between monitored / unmonitored and the resulting pile up of redundant squeue invocations which is consistent with my initial analysis.
>
> I don't remember seeing anything about redundant invocations. Is this written down somewhere? AFAICT the squeue goroutine can't run more than one squeue at a time.
>
> I see I wrote "squeue" in the commit message about simplifying exec.Cmd usage, but "sbatch" was the thing I actually fixed. So... did you also really mean "sbatch" here?

Yes, I meant sbatch. I have to review the logs, but I believe it was getting into a situation of running overlapping sbatch commands for the same container. Which would explain the FD leak if there was a bunch of stuck goroutines with pipes open.

- 10704 (sbatch loops too fast) → ae5eb12 10704: Rate-limit startup attempts per container.

> IIRC the Go "Ticker" sends a heartbeat every N seconds so for N=10 and it takes 9 seconds to do something, you'll only get 1 second of sleep before getting another tick. So squeue could still get called twice in < PollPeriod. Suggest MinRetryPeriod should have the same default as PollPeriod, instead of being disabled by default.
>
> MinRetryPeriod only affects how soon a container will be resubmitted with sbatch after disappearing from the slurm queue, and doesn't use a ticker.

Same confusion, I meant sbatch. I meant that if the intended default behavior is to rate-limit sbatch invocations to at least PollPeriod, with throttling disabled the actual time between two invocations could be less than PollPeriod because it could be run at the end of the previous period and again at the start of the next period. Setting throttle == PollPeriod (instead of 0) ensures that PollPeriod is the floor instead of the ceiling.

Rest LGTM.

#16 - 01/31/2017 08:31 PM - Tom Clegg

Peter Amstutz wrote:

> Yes, I meant sbatch. I have to review the logs, but I believe it was getting into a situation of running overlapping sbatch commands for the same container. Which would explain the FD leak if there was a bunch of stuck goroutines with pipes open.

OK. I looked into this on #10701. I didn't find that sbatch was getting stuck. It was possible for monitorSubmitOrCancel() to return before it even had a chance to check the slurm queue, though.

> MinRetryPeriod only affects how soon a container will be resubmitted with sbatch after disappearing from the slurm queue, and doesn't use a ticker.
>
> Same confusion, I meant sbatch. I meant that if the intended default behavior is to rate-limit sbatch invocations to at least PollPeriod, with throttling disabled the actual time between two invocations could be less than PollPeriod because it could be run at the end of the previous period and again at the start of the next period. Setting throttle == PollPeriod (instead of 0) ensures that PollPeriod is the floor instead of the ceiling.

Ah.

The intended default behavior is merely to behave correctly, i.e., avoid overlapping sbatch calls. In the default case, MinRetryPeriod is 0 and there is no effort (or need, afaik) to enforce a minimum time between two consecutive attempts. PollPeriod just establishes a minimum *average* time between attempts.

If you do want a minimum interval between sbatch attempts, you can use MinRetryPeriod.

#17 - 01/31/2017 09:24 PM - Peter Amstutz

LGTM.

#18 - 01/31/2017 09:40 PM - Tom Clegg

- Status changed from In Progress to Resolved

Applied in changeset arvados|commit:c3d4f8a585202ec58df5506934b698039c200b68.

Files

current	810 KB	12/09/2016	Peter Amstutz