Arvados - Feature #7399

[Crunch] [UX] Improve log throttling

09/29/2015 07:14 PM - Bryan Cosca

Status:	Resolved	**Start date:**	09/29/2015
Priority:	Normal	**Due date:**	
Assigned To:	Radhika Chippada	**% Done:**	100%
Category:	Crunch	**Estimated time:**	0.00 hour
Target version:	2016-03-02 sprint		

Description

Bug report

With the addition of log throttling (Exceeded rate 65536 bytes per 60 seconds (crunch_log_throttle_bytes). Logging will be silenced for the next 16 seconds.)

I can lose a lot of useful logs in that time window. My theory is to reduce the logging of crunchstat in order to see more what is going on in my computation without that clutter. If logging crunchstat does not add much to the rate, it might be useful to have a way to increase the crunch_log_throttle_bytes from an SDK or something.

Basically, I want a feature to be able to see all the log output without throttling (but I know that was impossible before crunch_log_throttle)

Implementation

Log throttling only applies to live logs, not the logs stored in Keep. Improve the message logs when jobs get throttled to help clarify this.

- Before "line.replace message" in source:services/api/lib/crunch_dispatch.rb#L478:

- [redacted]"
  ```
   A complete log is still being written to Keep, and will be available when the job finishes.\n
  "
  ```

- Make sure message always ends up with exactly one "\n" (currently there's a bit of inconsistency here).

When the log throttle is open and crunch-dispatch reads a log line that has a crunch-job "stderr" tag and content that **begins and ends** with "[...]" -- i.e., is the middle of a long line that has been split up by crunchstat -- handle it specially:

- If no other line matching this description has been passed through for 5 seconds, update the "last non-terminal line segment propagated" timestamp for the job, and handle this line normally using the existing log throttle code.
- If a non-terminal segment *has* been passed through in the last 5 seconds:
  - suppress this line and don't count it toward the throttle counters.
  - if this is the first time skipping a non-terminal line segment, replace the log line with an appropriate message and append the "complete log" reassurance above, as with the other messages.

    - `Rate-limiting partial segments of long lines to one every 5 seconds.`

"5" should be in a new config var Rails.configuration.crunch_log_partial_line_throttle_period.

Subtasks:

Task # 8472: Review branch: 7399-log-throttling	**Resolved**

Related issues:

Related to Arvados - Story #8552: [Crunch] Log throttling shouldn't throttle ...	**New**

Associated revisions

Revision cac5db66 - 03/02/2016 06:22 PM - Radhika Chippada

closes #7399

Merge branch '7399-log-throttling'

History

#1 - 09/30/2015 01:33 PM - Brett Smith

- Subject changed from Never silence logging to [Crunch] Never silence logging

- Category set to Crunch

Bryan,

Can you give some examples of jobs that have very verbose logging that you find useful?

I'm not sure we'll want to let users override the throttle completely, since the point of it is to protect the health of the cluster for all of its users, and an override undermines that purpose. But there might be other middle ground solutions we can consider. Throttling crunchstat logs before we throttle the job's stderr is definitely one good idea. Another thing we might do is stop writing the logs once they start coming in too fast, but hold on to the last few megabytes of them, and write out what's most recent if the job fails. Would that second one meet your needs? Basically, I'm wondering if there's are smarter ways to classify which logs are more likely to be important, and make sure those are saved for posterity, rather than just relaxing the throttle.

#2 - 09/30/2015 06:55 PM - Bryan Cosca

> Another thing we might do is stop writing the logs once they start coming in too fast, but hold on to the last few megabytes of them, and write out what's most recent if the job fails.

I really like that idea. My example is here:
https://workbench.tb05z.arvadosapi.com/collections/6686529fb30a9668cc9579a7d04faae3+91/tb05z-8i9sb-6to48rxvdualswg.log.txt

I used a tool and in order to have a better sense of what is going on and debugging, I needed to use an option to output a status bar (I think this is the most verbose way of debugging the tool). The status bar outputs at a very high rate. I do wish I could throttle this output bar, and I just want the "other" text that gets outputted. So the way you described it, it would be nice to stop writing the logs when they come too fast (progress bar) but I do want to hold on to the last few megabytes **in case** the error message lies in there. If the error is not in there, then its OK to throw out.

#3 - 11/16/2015 09:56 PM - Brett Smith

Check if the full logs made it to Keep. If so, check to see if there's something about the job that makes it difficult to find these logs (besides stuff like #7753).

#4 - 11/17/2015 02:19 PM - Brett Smith

- Target version set to Arvados Future Sprints

#5 - 11/17/2015 02:24 PM - Brett Smith

The log in Keep is complete. A lot of status bar lines have the [...] suffix/prefix, but if I'm following right that just means they've been broken across multiple lines, not that anything's been omitted.

#6 - 11/17/2015 02:41 PM - Brett Smith

Implementation notes:

- Clarify the message about what's being throttled (live logs only, not stored logs)
- Proactive throttle progress bars so more interesting logs are more likely to make it to the live log

#7 - 11/17/2015 02:57 PM - Brett Smith

- Subject changed from [Crunch] Never silence logging to [Crunch] [UX] Improve log throttling

- Description updated

#8 - 11/17/2015 07:04 PM - Tom Clegg

Omitting all but the first and last part of a split "epic log line" from the live log seems fairly reasonable, and easy.

Throttling (parts other than the first and last segments of) long lines to one per second or so would be better (no "looked totally silent, but was really just missing the \n" problem) although less trivial to implement.

#9 - 11/17/2015 08:49 PM - Tom Clegg

- Description updated

- Story points set to 1.0

#10 - 11/17/2015 08:49 PM - Tom Clegg

- Description updated

#11 - 11/30/2015 02:22 PM - Joshua Randall

This issue has just bitten me as well. My crunch_script had a task 0 that printed a number of warning lines (about files in the input collection that were being ignored), which triggered the log throttling. During the 52s of silenced logs, all of the other tasks got created, and I missed out on all the logging of which task was mapped to which host.

I guess I could raise the limit - I guess I understand the need for a limit, but 64K/m seems low (especially when applied instantaneously - i.e. I might not have an issue with 640K/10m).

Perhaps the throttling could implement some of the more clever log reduction strategies (things like identifying "similar" lines and replacing the actual printing of them with something like "Skipping N additional lines like this"), and also have different thresholds to apply over intervals (i.e. there might be an instantaneous rate that one might think is too much - maybe sthg like 100kB/s or 1MB/s - and also aggregate limits over longer averaged times (maybe 10MB/10m)?

#12 - 02/17/2016 07:49 PM - Brett Smith

- Target version changed from Arvados Future Sprints to 2016-03-02 sprint

#13 - 02/17/2016 07:56 PM - Brett Smith

- Assigned To set to Radhika Chippada

#14 - 02/18/2016 06:43 PM - Tom Clegg

- Description updated

#15 - 02/24/2016 12:35 PM - Radhika Chippada

- Status changed from New to In Progress

#16 - 02/24/2016 02:48 PM - Radhika Chippada

Branch 7399-log-throttling at f5c622f3018c7282375f5a0dca26e64c9ea8d982

- Added partial line throttling with the configured interval
- The description says: when log throttle is open and crunch-dispatch reads a log line that has a crunch-job "stderr" tag ...
  - It seems that the rate_limit method is invoked only for stderr log lines already. Do I need to some specific check for this? If so, please clarify.
- The description says: and content that begins and ends with "[...]" ...
  - I am checking if the line starts with and ends with "[...]". Please correct me if this is wrong interpretation.
- I couldn't find a way to add a test for this. Do we have any other rate limiting throttling related tests?

#17 - 02/25/2016 03:35 PM - Peter Amstutz

Radhika Chippada wrote:

> Branch 7399-log-throttling at f5c622f3018c7282375f5a0dca26e64c9ea8d982
>
> - Added partial line throttling with the configured interval
> - The description says: when log throttle is open and crunch-dispatch reads a log line that has a crunch-job "stderr" tag ...
>   - It seems that the rate_limit method is invoked only for stderr log lines already. Do I need to some specific check for this? If so, please clarify.

Normal output received from crunch-job has a header with (job uuid, pid, task) then "stderr" and then the output of the command.

```
4xphq-8i9sb-w9ok2v2mqo38c68
 9380 0 stderr run-command: md5sum /keep/a3e8f74c6f101eae01fa08bfb4e49b3a+54/md5sum.txt > md5sum.txt
```

Crunchstat lines can be recognized with an additional leading "crunchstat:"

```
2016-02-24_07:53:08.42422 4xphq-8i9sb-w9ok2v2mqo38c68
 9380 0 stderr crunchstat: cpu 0.0000 user 0.0000 sys 8 cpus
```

However re-reading the description, it doesn't say to treat crunchstat lines separately. (It mentions that crunchstat is responsible for long line breaking but that detail isn't relevant for throttling).

- The description says: and content that begins and ends with "[...]" ...

- I am checking if the line starts with and ends with "[...]". Please correct me if this is wrong interpretation.

You need to remove the line header up to "stderr" before looking for [...]

It appears that skip_counts (set when the middle part of a long line should be skipped) isn't preventing the line from being logged (rate_limit returns false).

- I couldn't find a way to add a test for this. Do we have any other rate limiting throttling related tests?

There are some tests in "test/unit/crunch_dispatch_test.rb" and "test/integration/crunch_dispatch_test.rb" but neither of those test logging. I suggest a narrow unit test around #rate_limit is probably going to be the most productive way to test it.

#18 - 02/29/2016 07:55 PM - Tom Clegg

I don't think we should creep scope here to treat crunchstat lines differently than other stderr. Both kinds of stderr are capable of getting "too busy", so it's not clear we even want to.

#19 - 02/29/2016 07:57 PM - Brett Smith

Tom Clegg wrote:

> I don't think we should creep scope here to treat crunchstat lines differently than other stderr. Both kinds of stderr are capable of getting "too busy", so it's not clear we even want to.

That part got split out into a separate story, #8552, so the desirability can be discussed there. It's out of scope for this story.

#20 - 02/29/2016 11:52 PM - Radhika Chippada

Corrected to return false when skipping partial line segment as pointed out in note 17, and added test.

#21 - 03/02/2016 04:02 PM - Peter Amstutz

line 450:

```
      line_splits = line.split('stderr ')
```

This would be better as:

```
      line_splits = line.split(' stderr ', 2)
```

You need to split on stderr as a standalone word (so it won't accidentally match "stderr" at the end of a job UUID) and limits it to two splits (because otherwise it could match "stderr" in the actual output).

However this still isn't ideal because it could still match output from other of a logging line that includes "stderr" in the text but isn't stderr from the job.

Here's a simple regex I have used to parse crunch-job lines: ^\S+ \S+ \d+ \d+ stderr (.*)
the content you're interested will be in the first capture group.

#22 - 03/02/2016 06:25 PM - Radhika Chippada

- Status changed from In Progress to Resolved

- % Done changed from 0 to 100

Applied in changeset arvados|commit:cac5db66cba0d5dd97c8434853bcbf2ab19fbda5.