Incident documentation/20151015-EventLogging

< Incident documentation

Summary

On Wednesday, 14 October, 2015 at 06:48 UTC, the Event Logging mysql consumer stopped inserting events. We would not have noticed except other alarms about raw vs. valid count led us to look at the dashboard, where we saw the eventlogging.overall.insertAttempted.rate dropped to zero.

Investigation and Recovery

Andrew found the mysql consumer died with this stack trace:https://gist.github.com/ottomata/39aa0f4b3dda1a60ed49. The mysql consumer was restarted at 15:03 UTC and started reading from kafka, recovering. However, it had a lot of back-pressure built up and used up a lot of resources while it was trying to catch up. It went back to normal around 20:20 UTC and has been normal since then. For about one hour after the mysql consumer restart, the difference between raw and valid events spiked up to over 4800, and then went back to normal. I'm not sure why that was affected.

Data Loss Report

I looked into three schemas that are usually pretty consistently getting events every few seconds: Edit, NavigationTiming, and ServerSideAccountCreation. All three of these schemas look like they're missing data, but not completely. There's a much lower than normal amount of data from 06:00 UTC to about 17:00 UTC. This means that when the consumer started back up it tried to recover but could not. We will try to save the data from Kafka and backfill manually. For future reference, this is how we grabbed the events that we needed to backfill:

```
ionice kafkacat -o 48850000 -c 1150000 -t eventlogging-valid-mixed -b kafka1012:9092 > recover.el.log
# NOTE: This, very weirdly, only got about 2 hours of data, even though the offsets used separately should span from 05:50 to 22:00.  Still looking into it.
```

Code Changes

The root cause is a timeout that killed the mysql consumer. That should be looked into more, but in the meantime we can make the mysql consumer take things slower when it tries to catch up. We think it tried to consume everything and ran out of memory, dying several times as it was catching up. We're tracking that and the fix here: https://phabricator.wikimedia.org/T115667

Actionables

- Status: 🟩 **Done** Look for holes in a few different schemas in the database and update this report.
- Status: 🟩 **Done** Consume kafka topic to a file and backfill manually (we ended up just using the all-events backup files).
- Status: 🟥 **Pending** Is there a way to make pykafka's timeout more robust? Or to automatically recover from this type of

problem?

Last edited on 26 October 2015, at 15:37