链滴

ROTATING GC LOG FILES

作者：fengjx

原文链接：https://ld246.com/article/1502711316726

来源网站：链滴

Garbage Collection logs are essential artifacts to optimize application’ s performance and tr uble shoot complex memory problems. Garbage Collection logs can be generated in a particu ar file path by passing the “-Xloggc” JVM argument.

Example: -Xloggc:/home/GCEASY/gc.log

But the challenge to this approach is: whenever the application is restarted, old GC log file will be over-ridden by the new GC log file as the file path is same (i.e. /home/GCEASY/gc.log).

Thus you wouldn’ t be able to analyze the old GC logs that existed before restarting the appl cation. Especially if the application has crashed or had certain performance problems then, yo need old GC Logs for analysis.

Because of the heat of the production problem, most of the time, IT/DevOps team forgets to ack up the old GC log file; A classic problem that happens again & again, that we all are famili r :-). Most human errors can be mitigated through automation and this problem is no excepti n to it.

A simple strategy to mitigate this challenge is to write new GC log contents in a different file l cation. In this article 2 different strategies to do that are shared with you:

1. Suffix timestamp to GC Log file

If you can suffix the GC log file with the time stamp at which the JVM was restarted then, GC og file locations will become unique. Then new GC logs will not over-ride the old GC logs. It c n be achieved as shown below:

UNIX:

```
-XX:+PrintGCDetails -XX:+PrintGCDateStamps -Xloggc:/home/GCEASY/gc-$(date +%Y_%m_
d-%H_%M).log
```

WINDOWS:

```
# Generating time stamp & replacing '' with 0 and / with -
set file_suffix=%DATE:~-4%-%DATE:~4,2%-%DATE:~7,2%_%TIME:~0,2%-%TIME:~3,2%-%TIM
:~6,2%
set file_suffix=%file_suffix: =0%
set file_suffix=%file_suffix:/=-%

set JAVA_OPTS=%JAVA_OPTS% -XX:+PrintGCDetails -XX:+PrintGCDateStamps -Xloggc:c:\wor
space\newUploads\gc-%file_suffix%.log
```

This strategy has one minor drawback:

a. Growing file size

Suppose if you don’ t restart your JVMs, then GC log file size can be growing to huge size. B cause in this strategy new GC log files are created only when you restart the JVM. But this is n t a major concern in my opinion, because one GC event only occupies few bytes. So typically C log file size will not grow beyond a manageable point.

2. Use -XX:+UseGCLogFileRotation

Another approach is to use the JVM system properties:

-XX:+PrintGCDetails -XX:+PrintGCDateStamps -Xloggc:/home/GCEASY/gc.log -XX:+UseGCLo
FileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=2M

When '-XX:-UseGCLogFileRotation' is passed GC log rotation is enabled by the JVM itself.

'-XX:NumberOfGClogFiles' sets the number of files to use when rotating logs, must be >= 1. The rotated log files will use the following naming scheme, .0, .1, ..., .n-1.

'-XX:GCLogFileSize' defines the size of the log file at which point the log will be rotated, m st be >= 8K

But this strategy has few challenges:

a. Losing old GC Logs

Suppose if you had configured -XX:NumberOfGCLogFiles=5 then, over a period of time, 5 GC og files will be created:

gc.log.0 — oldest GC Log content

gc.log.1

gc.log.2

gc.log.3

gc.log.4 — latest GC Log content

Most recent GC log contents will be written to 'gc.log.4' and old GC log content will be pre ent in 'gc.log.0'.

When the application starts to generate more GC logs than the configured '-XX:NumberOf CLogFiles' in this case 5, then old GC log contents in gc.log.0 will be deleted. New GC events will be written to gc.log.0. It means you will end up not having all th generated GC logs. You will loose the full visibility of the events.

b. Mixed-up GC Logs

Suppose application has created 5 gc log files i.e.

gc.log.0

gc.log.1

gc.log.2

gc.log.3

gc.log.4

then, let's say you are restarting the application. Now new GC logs will be written to gc.log.0 file and old GC log content will be present in gc.log.1, gc.log.2, gc.log.3, gc.log.4 i.e.

gc.log.0 — GC log file content after restart

gc.log.1 — GC log file content before restart

gc.log.2 — GC log file content before restart

gc.log.3 — GC log file content before restart

gc.log.4 — GC log file content before restart

So your new GC log contents get mixed up with old GC logs. Thus to mitigate this problem y u might have to move all the old GC logs to a different folder before you restart the applicati n.

c. Forwarding GC logs to central location

In this approach, current active file to which GC logs are written is marked with extension ".c rrent". Example, if GC events are currently written to file 'gc.log.3' it would be named as: gc.log.3.**current** '.

If you want to forward GC logs from each server to a central location, then most DevOps engi eers uses 'rsyslog'. However this file naming convention poses significant challenge to use 'rsyslog', as described in this blog.

d. Tooling

Now to analyze the GC log file using the GC tools such as (gceasy.io, GCViewer....), you will ha e to upload multiple GC log files instead of just one single GC Log file.

Conclusion

You can debate on which approach you want to take for rotating GC log files, but don' t deb te on whether to rotate the GC log files or not. It will come very handy when need comes. You never know when need comes.

原文连接：https://blog.gceasy.io/2016/11/15/rotating-gc-log-files/