

文章

[Hao Ma](#) · 一月 10, 2021 阅读大约需 6 分钟

我的日志不断增长 - 如何将其减到最小

在我发了[前一个帖子](#)后，有人催促我说重点 - 好吧，于是我找到了我的“明星”日志 global，也就是那些最占空间的 global - 但如何避免这种情况呢？如何最大程度地减小日志？

[免责声明：有些人在看到本帖后可能仍然会失望 😞，那就等下个帖子吧...]

不过很遗憾，你还得多失望一会儿，我需要推迟讨论（不会很久...），因为我要先提出两个我认为很重要的问题，之后再讨论如何避免记录那些我们不需要的日志，这两个问题是：

1. 到底有哪些内容会被写入日志？
2. 为什么？

我认为在讨论不记录日志之前回答这两个问题非常重要的原因是，从根本上来说，日志记录是有好处的，所以数据首先会放在日志中，如果你决定不记录日志，必须有非常充分的理由。

让我们尝试回答这两个问题 -

通常每个 SET 和 KILL 命令（以及 TSTART 和 TCOMMIT/TROLLBACK）都会记录在日志文件中。

例如，如果我运行以下命令

我会在日志中看到：

日志文件有几个用途：

- a. 为了防止在系统崩溃后的启动恢复中发生数据丢失 - 当系统启动时，如果日志中的操作尚未记入数据库，系统会前滚这些操作（假设它们不是未提交事务的一部分，否则它们会被回滚）。
- b. 作为灾难恢复计划的补充 - 允许恢复自上一个可用备份以来更改的数据。
例如，如果在昨天夜间运行了一次备份，由于某种原因，数据库（即 CACHE.DAT）文件在中午时受损（磁盘损坏或其他原因），那么可以从备份中恢复，让状态回到早上的样子，然后从日志恢复到中午发生系统故障之前的点，以将数据损失降到最低。
- c. 事务回滚支持 - 无论出于何种原因，都能回滚事务。

更多信息，请参见[此处](#)。

（写一个单独的帖子可以更详细地介绍每个用途，并说明其用法。还可以介绍日志中的回滚是什么样子等等。这不是我目前正在计划的帖子，但欢迎其他人阐述这个主题。

所以记录数据日志是有充分理由的，它非常有用，甚至至关重要（重要到甚至有一个开关的说明是，如果日志记录因

某种原因失败，我们应该冻结整个系统)。

(希望此时不会让你因为考虑不记录日志而感到内疚...)

Caché 中的每个数据库都有一个指示数据是否被记录日志的标志。默认和推荐的设置是记录日志。即使你决定不记录，你也应该意识到，事务中的任何 SET 或 KILL 仍然会被记入日志，即使数据所在的数据库被标记为不记录日志（为了允许上文所述的回滚）。[一个重要的例外是位于 CACHETEMP 数据库中的 global，我们稍后会用到它]

例如，SAMPLES 数据库未记录日志，因此简单的 SET 不会被记录，但事务中的 SET 会被记录：

日志不显示第一个 SET（在事务外部），只显示第二个：

对于事务内自动保存的对象同样如此。

再以 SAMPLES 为例：

我们得到这样的结果（尽管我们没有明确发出 TSTART 命令，但对对象归档器会在后台执行此操作）：

这样，虽然简短，但我们的 2 个问题（日志记录什么和为什么记录）都得到了回答。

这将引导我们进入下一步 - 如何避免某些数据被记入日志。

你会意识到的第一个问题是决定是否可以真正“承担得起”不记录那些你想要避免填满日志的数据。

不过让我先说一个非常简单的案例 - “哇！我不知道我们在设置这个 global ？！”

例如，查看日志可以发现，也许在代码中的某个地方，我们设置了一个临时开关，可能是设置某个时间，或者是为了调试目的而需要写入一些日志，或者是具有临时调查性质的某种机制，而我们完全不需要这个开关，我们可以删除该代码（或者注释掉，或者不调用，或者关闭开关等等）。

这是一个相对容易的案例，我们不需要继续讨论是否以及如何不记录日志。

现在，关于此案例，你可能会想 - 你需要看日志才能发现这个“失控”的 global 设置吗，你看不到数据库实际在增长吗？

好吧，也许的确如此，数据库也确实在增长，但你忽略了，或者没有给予过多关注并且/或者认为这只是自然增长。但也可能是另一种情况，这是我要强调的一点，就是日志增长不一定与数据库增长相关联。

举一个极端的例子：

你可以设置一个 global 的值，让其在两个值（比如 0 和 1）之间交替，每秒多次。数据库完全不会增长，因为数据的大小没有变化，但日志中会有成千上万的 SET 条目。

或者你可以设置一个临时的大型 global 树结构，数秒钟后将其终止，那么树结构占用的数据库大小差不多是恒定的，但是日志会因为记录所有 set 和 kill 操作而填满。

假设情况并非如此，你对设置此数据并不感到惊讶，实际上你很清楚这一点，并且你的应用程序需要设置此数据。

那样的话，我们又回到了前面的问题 - 你是否真正承担得起不记录这些数据的日志？

为了回答这个问题，我们试着把它分成几个小一点的问题：

1. 你能承受在崩溃前输入（或删除）的数据在崩溃后不存在（或重新出现）吗？
2. 你能承受所输入（或删除）的数据在系统恢复过程中无法恢复（或重新出现）吗？
3. 你不需要这些数据作为事务的一部分被回滚吗？

如果你对全部 3 个问题的回答都是“是”，那么你可以考虑不记录这些数据的日志了。

但是如果你需要此数据在崩溃后恢复，或者作为备份的一部分恢复，或者需要作为事务的一部分回滚。你就需要记录日志。

在这种情况下，你只需要计算并确定真正的日志增长率，并确保有足够的磁盘空间

[你可以看我的另一个[帖子](#)，一些相关代码在 Git 上，可以帮助估计 Ensemble 接口所需的日志空间。当然你也可以修改这些代码来用于其他用例]。

以下是我将在下一个帖子介绍的选项，先让你们预览一下：

- 关闭整个系统的日志记录
- 将 global 映射到 CACHETEMP
- 将 global 映射到不记录日志的数据库
- 关闭对进程的日志记录
- 关闭用于对象归档的事务

下次再见...（我保证到时候说重点...）

这是[下一个帖子](#)。

[#日志](#) [#系统管理](#) [#Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E6%88%91%E7%9A%84%E6%97%A5%E5%BF%97%E4%B8%8D%E6%96%AD%E5%A2%9E%E9%95%BF-%E5%A6%82%E4%BD%95%E5%B0%86%E5%85%B6%E5%87%8F%E5%88%B0%E6%9C%80%E5%B0%8F>