GammaLib - Change request #1817

GLog should regularly update log file

07/18/2016 02:08 PM - Mayer Michael

Status:	New	Start date:	07/18/2016
Priority:	Normal	Due date:	
Assigned To:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

I stumbled upon this when sending ct...-jobs to the batch system. In order to monitor the progress and sanity of the tools' work, I wanted to monitor the log files. Unfortunately, the log file gets only filled when the job is done and the ctool destructor is called. My proposal is to change this logic such that the log file is updated regularly.

1. Either we use the command GLog::flush in the logger (e.g. after a newline is requested)
2. Or we use GLog::flush in each ctool after certain tasks have been performed.

History

#1 - 07/18/2016 05:12 PM - Knödlseder Jürgen

GLog has a member m_max_length that defines the length of the buffer. Only if the buffer length is reached it is written to disk. We could simply make m_max_length smaller. The goal of having a buffer is to avoid excessive I/O when a lot of information is written to the log file.

For the moment the buffer length is 8192 characters, maybe setting it to 256 (corresponding to a bunch of lines) would be a reasonable compromise?

#2 - 07/19/2016 01:09 PM - Mayer Michael

I just made a quick check and changed m_max_length locally to 256 (and also tried 64). However, this did not change the behaviour that the log file gets only written after the job has finished. Can you observe the same thing, or maybe is it just a feature of my batch farm?

#3 - 07/22/2016 09:01 AM - Knödlseder Jürgen

user#77 wrote:

> I just made a quick check and changed m_max_length locally to 256 (and also tried 64). However, this did not change the behaviour that the log file gets only written after the job has finished. Can you observe the same thing, or maybe is it just a feature of my batch farm?

I have to look into that, but I'm a bit surprised that you don't see the change. I remember to have watched log files using

tail -f ctlike.log

and have seen updates as the job was proceeding. So maybe indeed a feature of the batch farm (copying files from scratch space to your user space?)

#4 - 07/22/2016 09:45 AM - Mayer Michael

So maybe indeed a feature of the batch farm (copying files from scratch space to your user space?)

I have just tried running a ctool on command line on my mac and monitored the log-file from another console. With the reduced value of m_max_length=256, the log file gets indeed written after ~40-50 lines. Probably there is a feature of my bath farm - i will investigate (I don't think it is copying the file to the user space, since the log file is present from the start, it is just empty)

In particular for tools like ctlike where one might want to know details about the fit process (iterations), it might be worthwhile to even reduce that number further. We could for instance think about adding a default parameter to all the tools that specifies after how many lines the log file will be updated. Or would this be too complex and confusing for the user?
Alternatively, each tool could set the m_max_length parameter in its GLog-instance independently (maybe even taking into account the chatter in order to avoid too many file open/close actions?)
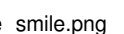
#5 - 07/22/2016 11:07 AM - Mayer Michael

Interesting: I have tested the same thing on the SL6 at the Zeuthen computing center (two terminals, on runs the tool the other for monitoring the log file). The result is that it seems on SL6 the log file doesn't get filled until the job is finished (regardless of m_max_length). I have no idea what could cause this. Does anyone have access to a SL6 system to reproduce this?

#6 - 07/22/2016 11:31 AM - Knödlseder Jürgen

I will check.

#7 - 08/05/2016 03:04 PM - Knödlseder Jürgen

It turned out that the kernel has an internal buffering algorithm, so I probably overdid things by using my own buffer. Anyways, now it's there smile.png

I added a statement

```
// And now flush the kernel buffer
std::fflush(m_file);
```

to the GLog::flush() method which flushes the kernel buffer.

#8 - 08/08/2016 11:46 AM - Mayer Michael

Thanks, but honestly, I still don't see a difference smile.png