20170424 ARC-CE Crisis meeting minutes

Present: Oxana, Florido, Maiken, Aleksandr, David, Dmytro, Jens, Andrej, Gianfranco

Andrej introduces the main discussion points: sites are experiencing problems with A-REX. Some old and known, some new and difficult to debug.
Two typical scenarios:

A. A-REX works fairly well when
    - There are no or not many transfers, i.e.
        i. Running only pilot jobs
        ii. Running jobs with lightweight data staging (eg BOINC).
B. A-REX starts having issues when there is lots of **datastaging** ongoing. Typical malfunctions are:
    - hangs / crashes and requires manual restart
    - Huge CPU usage (400% in some cases)
    - High memory load for O(1000) jobs, anonymous memory paging up to 7GB in some cases
    - When ATLAS increased transfer activity to the order of 100K input files per day it clearly killed A-REX performance
    - The data delivery seems to be reliable in terms of transfer performance but not in the way it handles multiple transfers
    - Switching to ARC 5.3 made everything worse to the point that many downgraded, and now the performance with 5.2 is acceptable

Meeting scope: To identify how to proceed in investigating all these issues.

Jens adds that there are some issues related to the scheduling of parallel transfers, he can reproduce that in 5 mins, he calls it the “Thread Congestion Problem”

Aleksandr laments that it is sometimes hard for the developer to get the information required even if the bug is being tracked in bugzilla.

Dmytro reports that Ulf sent a review of the delegation handling with suggestions on how to improve it.

ACTION: Ulf should create a bugzilla entry with these comments and suggestion.

It comes out from the discussion that there are problems at different levels, to be addressed in different ways, for which ACTIONS will be taken:

1. A-REX crashing
2. A-REX hanging
3. A-REX/datastaging waiting for some threads to finish or one thread being extremely slow and others waiting on the lock. “Thread Congestion problem”
4. A more consistent way of gathering crash information must be established
5. Glibc issues that are hard to reproduce.

1 and 2:

Aleksandr said they can be tackled in two ways, either dissection (that is, testing A-REX for every commit incrementally from 5.2 to 5.3 and identify when the problem arises) or by reproducing in controlled environment.
David suspects that is the Openssl changes that are causing the issues, as he looked at the commits himself and found that the only critical code change is there.
Most of the audience agrees that the problem with crashes had been there way before 5.2. Andrej suggests it would be beneficial to use profiling tools.
Oxana says that this behaviour is typical of memory leaks, Aleksandr agrees.
ACTIONS:

- Maiken and Dmytro will arrange so that dissection from 5.2 to 5.3 can be carried on, with the support of Aleksandr. Aleksandr should be granted access on the target resources.
- Maiken will look into profiling tools
- Andrej will provide sample jobs or sample job loads that might cause the problem
- Dmytro, Jens and Andrej can provide Aleksandr with access to the clusters

3. Thread congestion problem

If the jobs are in need of datastaging up to 1K files, A-REX works fine.
As soon as the needed files go over 1K threads seems to end up in some mutex locking issue. A-REX does not crash nor hangs, just the threads are waiting for some slow thread to finish or release the lock.
Datatransfer throughput is not affected, the transfer speed is fine, it's the number of concurrent transfers that dramatically goes down, sometimes in the order of 40-70 transfers but in some cases even just 1.
David says that maybe some of the code that manages these threads should be reviewed and new scalable strategies should be found.
ACTIONS:

- David and Jens should document and report these issues because they have investigated them deeper. Could be done by some text that describes the issue within a bugzilla ticket.
- David thinks that one should investigate if it is possible to separate datastaging in a different process than A-REX
- Jens can provide Aleksandr with access to the clusters
- ARC developers: Longer-term re-architecture of data staging (current one is 7 years old)

4. A more consistent way of gathering crash information must be established

Andrej thinks that we should have a system within NG/NDGF to gather and collect backtraces. Dmytro is using some of the scripts now provided by ARC, and suggests

to create a script that gathers together relevant logfiles. As it would be nice to gather this information, Andrej wonders if it is doable with Logstash.
ACTIONS:

- Dmytro will provide the script and will investigate if Logstash can be used.

5. Glibc issues that are hard to reproduce

There is a closed bug about a Glibc spawn process issue causing hanging that happens randomly but cannot be reproduced on demand. It is not clear if this issue still exist, for example the C++ code that triggered that when respawning the infoproviders process has been changed now so it's not as visible as before. Jens shows that we are using deprecated thread handling in ARC. Aleksandr thinks that this is not causing the issues, and if we go that way we might drop compliance with thread handling prior to C++11
ACTIONS: All ARC developers: Evaluate if it is worth to update the code to the recommended thread handling api and how much this affects pre C++11 systems. Investigate if the problem is related to the number of threads spawned during the lifetime of A-REX.