Some questions raised by Akos FROHNER:

Dear all,

I would like to highlight a few points before the discussion on 'SRM busy':

1- What kind of flow control do we want to implement?

- Client receiving the busy signal shall retry the same operation later, preferably with an exponential timeout.

Is it worth implementing this for trivial (mkdir, stat) operations, where the bulk of the processing time is authentication and XML parsing, which has to happen before returning any SRM level message?

Is there a way to avoid authentication and SOAP processing to send this signal?

- If SRM requests are coming continuously, then delaying one will not help the overall situation. Client services with internal state (not lcg-util, but like FTS) could scale back their submission rate to help the overall situation.

How should they scale back? Halving the submission rate and then ramping up linearly is a option, however can be too aggressive. Is there a way to give more information, like SE has 40 transfer slots and 39 are used in the last hour? Is there a way to provide this information for others, like experiment data management frameworks that they could choose another SE for the transfer?

2 - Backward compatibility:

- The currently released versions of lcg-util will use exponential retry on srm-get and srm-put, if they return SRM_INTERNAL_ERROR. On the other hand with other operations (srm-ls, srm-mkdir) and with other error codes (SRM_FILE_BUSY) the transfer job would be aborted.

- The currently released versions of FTS will fail transfer jobs, in case of SRM_INTERNAL_ERROR or SRM_FILE_BUSY is received at request level. If it is received in the final srm-put-done, then FTS tries to clean up by removing the file. Although it will re-try the whole transfer job later.

As I see the roll-out time of new clients (lcg-util, FTS) is about 6 months, so even if we make a decision now, implement it in the clients, the servers cannot deploy the change before the summer, unless possibly breaking existing clients.

Is it possible to find a backward compatible solution, at least for the transitional period? For example a new field in srm-ping?

3- Modeling the situation

I have enumerated a few possibilities above and we have a lot of logs of various storage elements and client services. Would it be possible to have a look into those logs to support one or the other choice with numbers?

For example if your SE has 95% the requests coming from FTS, then making a backward incompatible change, which breaks FTS is not a good choice. Or if your SE is suffering from peak loads, then the simple exponential timeout would help.

FYI I try to track the FTS related items here: https://savannah.cern.ch/bugs/index.php?44018 ⧉

-- ElisaLanciotti - 09 Feb 2009

This topic: LCG > QuestionsAkos
Topic revision: r2 - 2009-02-10 - ElisaLanciotti

Some questions raised by Akos FROHNER: