

## Sites Feedbacks

- The algorithm could look at TOTAL bandwidth between SE1 and SE2 to make the decision of lowering or raising the NTransfers. That sounds even more natural to me. I understand why one would want to look at the average rate to optimize other aspects (as load in GFTPs per BW unit), but in this case, all we got was the optimization preventing the transfer to go faster.
  - ◆ The TOTAL bandwidth between SE1 and SE2 is shared between experiments and other activities it is not dedicated to a given VO. So making FTS trying always to saturate that bandwidth results in SEs overload and decrease of the efficiency.
- Site admins, operators et al being able to override the default or initial value of parallel transfers. It looks very inefficient to want to do fast a transfer, and wait some hours for the algorithm to run and ramp up to a decent number of transfers. For automated systems as PhEDEx this makes sense.
  - ◆ It is already possible to overwrite the default config.

## Ongoing tasks

- Optimizer algo. Test
- FTS3 monitoring
  - ◆ Retries
  - ◆ Files monitoring
  - ◆ Activities fairshare
  - ◆ Users files monitoring
    - ◇ in Job Monitoring Dashborad: done
    - ◇ in FTS3 Dashboard
- Enhancement
  - ◆ Make the proxy delegation with the REST client easier to use for a system managing several users.

## Optimizer recovery from failures

### Test scenario

1. Saturate the link CNAF > CERN using Phedex until reaching a stable number of active files
2. Simulate the storage failure
3. Check how much time is needed to reach the minimum number of actives
4. Simulate the storage recovery
5. Check how much time is needed to reach the maximum number of active files in 1)
6. Repeat 1 - 5 with optimizer aggressiveness set to 3 (default is 1)

During those steps, the parameters to check are the number of active files, Throughput, Efficiency and Total data transferred

### First round

- To simulate the storage failure in 2) we tried to remove the user permissions on the output directory/set the user quota on the storage to 0. Since they are considered as non recoverable error by FTS, the optimizer did not react which it does make sense. Consequently, we were not able to simulate 2)
- No particular correlation could be concluded between active files/throughput (see plots). So pushing more files does not mean increasing the throughput. We saw within the same number of active files the throughput could be too much different. There is no particular other traffic over that link that it is interfering. It could be due it to the local access? xrootd/fax? to check...

- To simulate the storage failure we need to set the timeout config. of the transfer over that link to a small number

-- HassenRiahi - 28 Jun 2014

---

This topic: LCG > FTS3Commissioning

Topic revision: r7 - 2014-07-24 - HassenRiahi



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback