FTS 3 Status

Zsolt Molnar
Zsolt.Molnar@cern.ch

CERN, IT-GT-DMS

17 October, 2011
3rd EMI All Hands Meeting, Padova

FTS 3 main goals (reminder)

- Solve scalability problems of channel model
- Solve configuration and management problems
- Solve software maintenance problems (architectural limit reached)
- Eliminate redundancies between data management projects

Related efforts

- GFAL 2 (Adrien)
  - In core of FTS 3
  - Transfer/control plugins, transfer logic, infosys
  - FTS 3 schedules, monitors, supervises LCG_Util transfers
  - Very good progress, convincing results
- GLUE 2 (Oliver, Michail)
  - GLUE2 support from the beginning
- Data consolidation effort
  - LCG_Util and additional transfer plugins (HTTP...) from ARC

Work done

- Finishing "old" projects
- Last FTS 2 release (2.2.8) - only critical future releases
- Last LCG_Util / gLite release (1.11.16-3)
- GFAL 2, GLUE 2
- Shared components between LCG_Util and FTS 3
  - *is_interface, gridftp_ifce, srm_ifce*
  - Released in LCG_Util 1.11.18 for EMI
  - First feedback and fixes in LCG_Util 1.11.19

Next six month plans

- Prepare GFAL 2 release
  - LCG_Util based on GFAL1 / EMI: keep releasing until GFAL2 is out
  - then critical fixes only
- Work out / research new FTS 3 concepts
  - Series of prototypes concentrating on individual FTS 3 aspects. Examples:
    - Transfer over *lcg_cr*
    - Scheduling based on simulated SE parameters
    - Java-less web service
    - *https://svnweb.cern.ch/trac/fts3*
    - More?
- Put together FTS3 prototype 1 using the results

SE-based scheduling - Goals

Develop *strategies*

- WHAT sorf of info FTS needs
  - Users
- HOW to deliver it to FTS
  - Admins: FTS and SE

Discissions with users and administrators

SE-based scheduling - WHAT

First proposal in wiki

https://svnweb.cern.ch/trac/fts3/wiki/Configuration

- Deducted from channel model (FTS 2)
- Performance comparable to channel model expected
- format
  - key:value pairs
  - `SE:<SE endpoint>:<property name>=<value>`
  - `SE:public_srm.cern.ch:TransferType=urlcopy`
- content
  - `TransferType - VOShare(<VO>) - SpaceTokenShare(<VO>, <space token>) - PublicShare - TransferProtocols - ControlProtocols`
  - add, remove, change what?

SE-based scheduling - HOW

1. Interactively.

- By config CLI: configure also remotely.
- PRO: No change in SE
- PRO: Available always, "supports" new developments immediately
- CON: Not scalable, like in channel era
  - But PRO: SE-s can be managed independently, no pair (channel) agreements
- Going to be supported: early phase, testing, etc.
- Authorized SE or FTS admins can feed config database.

SE-based scheduling - HOW

2. Automatically.

- Info retrieval plugins in FTS for each supported SE providers
  - PRO: Burden on FTS development only
  - PRO: Info already available
  - CON: Not precisely what FTS needs
  - CON: No SE transparency (idea of supported SE-s)
  - CON: Internal interface gets external
- SE-s publish their state - GocDB?
  - PRO: Standardized (EMI goal)
  - PRO: Open for other tools, use cases
  - CON: Burden on SE providers
    - Admins and developers must contribute

SE-based scheduling - HOW

3. **Autonomously.**
    - The coolest FTS 3 feature.
    - Start with good "first" values ...
        - hard-coded FTS defaults for each properties
        - ... or specify it interactively
        - ... or retrieve it automatically
    - ... then watch past transfers
    - ... analyze them ("learn")
    - ... adjust the parameters.