

FTS3 and GFAL2



Michal Simon, CERN

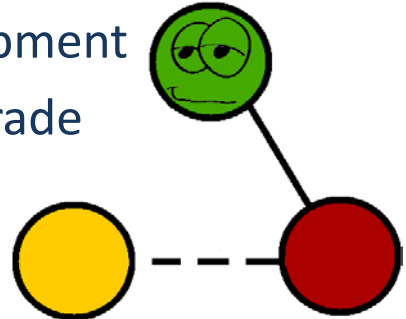
Manchester, 10/04/2013

- FTS3
 - Status
 - Features
 - Directions
- GFAL2
 - Introduction
 - Features
 - Summary

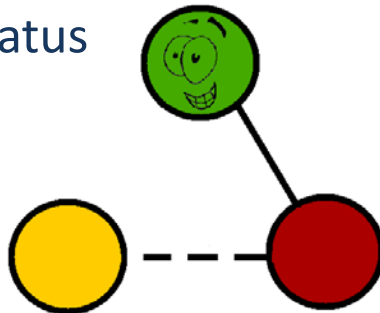
- Stable and functionally comparable to FTS2
- Installed at CERN (pilot), RAL, ASGC, PIC and BNL
- Used by CERN experiments (Atlas and CMS) for production and debug transfers
- Stress and scalability testing (vs. FTS2):
 - 26.6% less number of VMs
 - 14.2% less memory
 - 2.2% more transfers



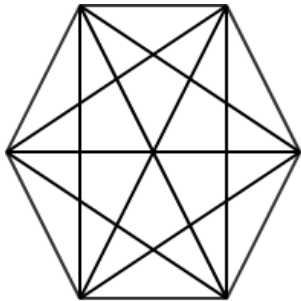
- Alternative Database back-ends (Oracle or MySQL)
- Additional protocols: xroot and http
 - FTS3 plugins are under development
 - Infrastructure requires an upgrade
- Session reuse (improved performance for the ‘many small files’ use case)



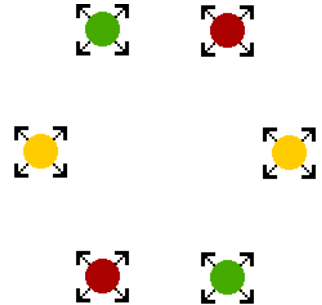
- Smart retry logic
- Improved UI:
 - FTS3 is backwards compatible with glite clients
 - There is a new, improved CLI that gives access to the new features (e.g. JSON formatted output)
 - Transfer submission and status retrieval APIs available:
 - *in Python*
 - *through REST interface*



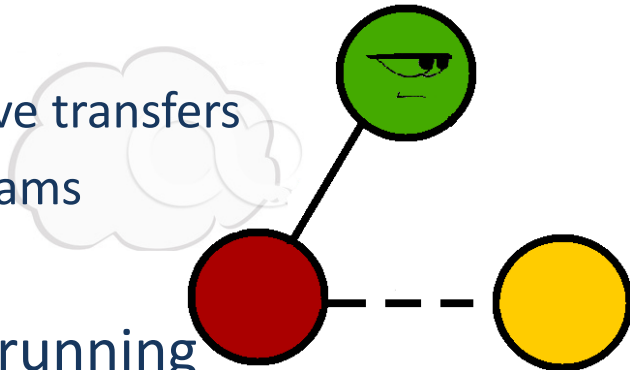
- Works out of the box (no configuration required):
 - *Good defaults*
 - *Auto-tuned transfers*
- Endpoint-centric manual configuration
 - *VO specific shares*



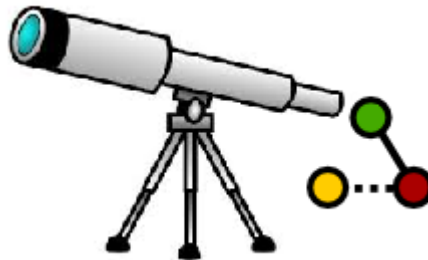
*Turns quadratic scaling
behaviour into linear*



- Dynamic adjustment (based on recent history) of:
 - Number of active transfers
 - Number of streams
 - The timeout
- FTS3 has been running for 8 months relying on the auto-tuner
- Planned for the future:
 - VO specific shares on top of the auto-tuner

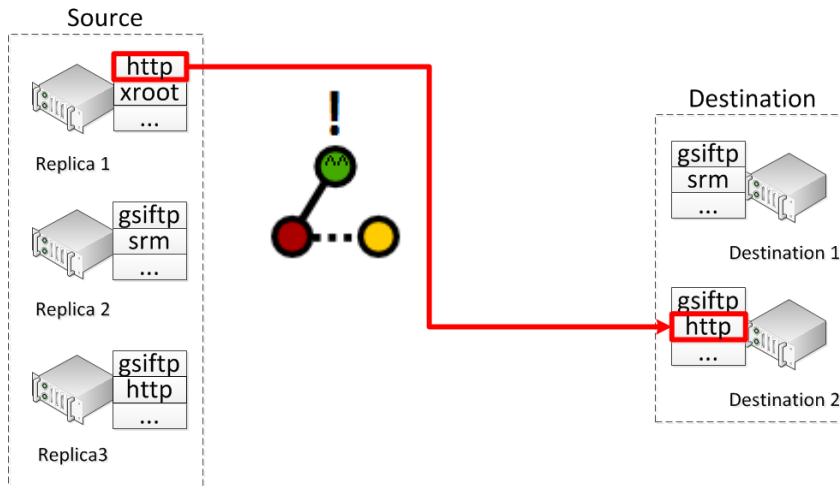


- FTS 3 publishes monitoring messages
 - per every state transition
 - periodically (state of the queue, failure rate, etc.)
- Web interface for monitoring:
 - transfer-jobs
 - queue
 - statistics
 - auto-tuner
- Dashboard monitoring



FTS3: Multiple replica utilization

- Source: alternative replicas (or protocol)
- Destination: alternative SEs (or protocol)
- FTS3 will choose the best SEs and the best protocol to carry out the transfer job



- Released in EMI3
- Forthcoming releases in EPEL
- Deployment plan: to be decided
 - *Either central or distributed*
 - *Can be kept open until mid 2014*
- FTS3 pilot: <https://fts3-pilot.cern.ch:8443>
(feel free to try it out!!!)
- Wiki: <https://svnweb.cern.ch/trac/fts3/wiki>
- Stress testing:
<https://svnweb.cern.ch/trac/fts3/wiki/StressTesting>

- One toolkit for all the GRID and Cloud data operations:
 - Client data access
 - *support for all common protocols and storage systems*
 - Data management
 - *unified API for all data management*
 - Data transfer
 - *3rd party transfer support*

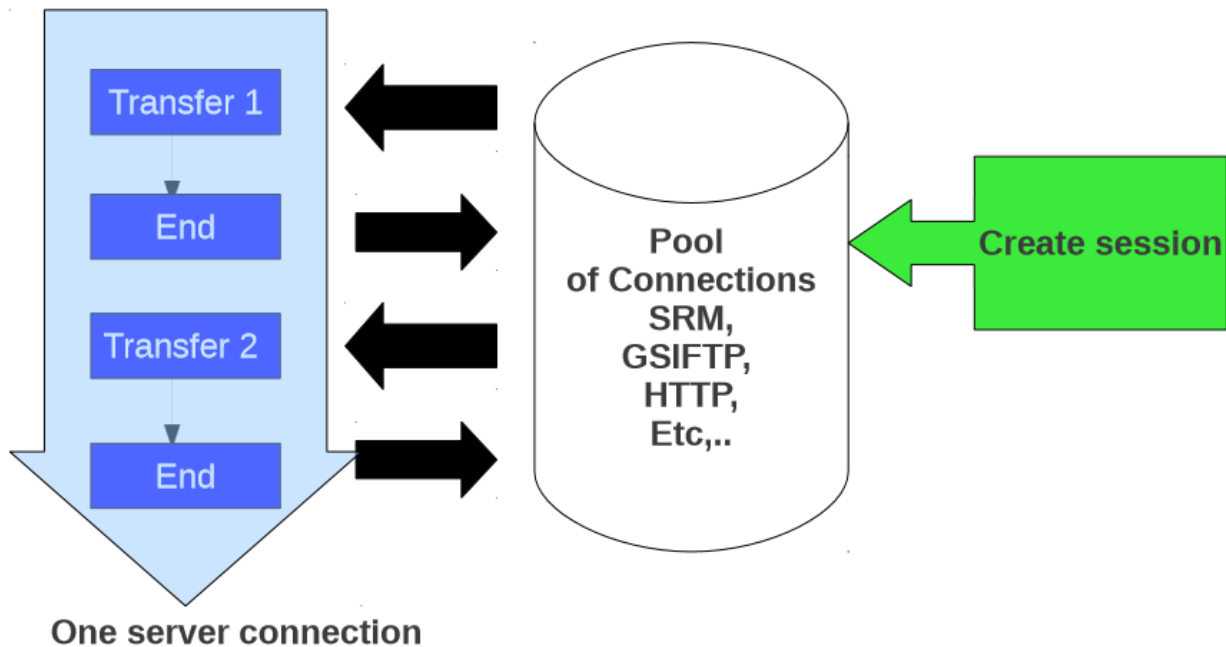


- Carries out all the transfer-jobs submitted to FTS3 (3rd party transfers)
- Hides the GRID complexity
- Provides common API for all the access and transfer protocols
- Thanks to the adopted plug-in mechanism reduces the number of dependencies



- Take *a* and *b* in:
 - *{srm, guid, rfio, dcap, gsidcap, gsiftp, file, xrootd, http, dav}*
- Automatic protocol resolution:
 - *gfal_copyfile(a://src/file, b://dest/file)*
- GFAL2 does all the work for you!

GFAL2: Thread-safe session reuse



- Supports all common storage elements
- Supports lots of protocols:
 - SRM, GRIDFTP, GUID, XROOTD, HTTP(S) and more
- Supports POSIX extended attributes
- No environment variables
 - although, it is compatible with the environment variables from GFAL 1.0 era for an easy transition

- libgfal2:
 - *a C library*
 - *A set of independent plugins*
- gfal2-python:
 - *simple to use python-bindings*
- gfalFS (fuse module for gfal2):
 - *Allows to mount any GRID endpoint in a local folder*
- gfal-tools:
 - *experimental CLI: gfal-ls, gfal-copy, gfal-mkdir etc.*

- Core component of FTS3
 - will be supported in any case
- Released in EMI2 and EMI3 (www.eu-emi.eu)
 - will be supported after the end of EMI
- Released in EPEL
- There are plans to do packaging for Debian