

New FTS: FTS 3

Zsolt Molnár

Zsolt.Molnar@cern.ch

CERN, IT-GT-DMS

11 July, 2011

WLCG Collaboration Workshop



File Transfer Service

- ▶ Data transfer **job scheduler**
- ▶ Submit → queue → execute when optimal
- ▶ CLI of web service → database → transfer agents
- ▶ Monitoring



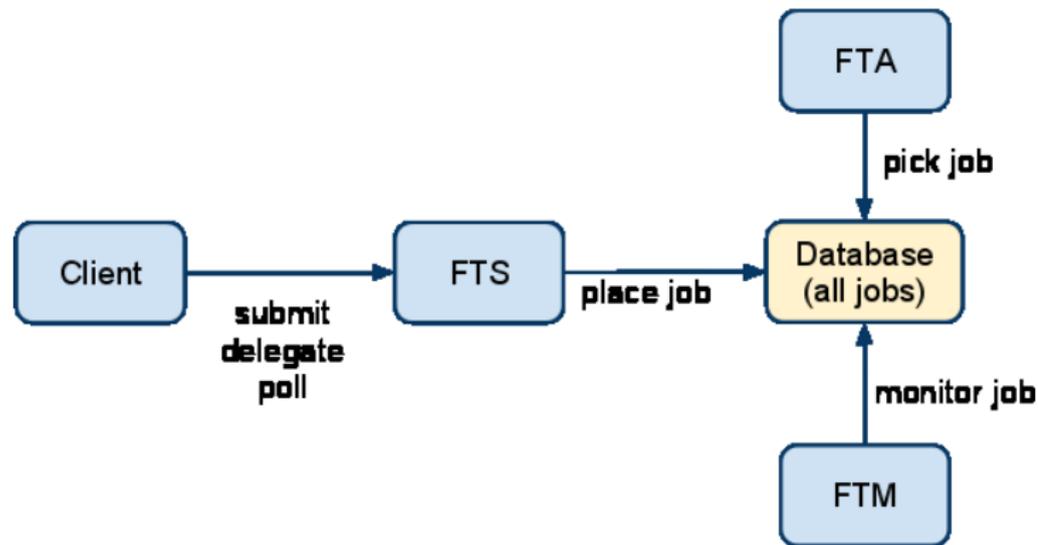
- ▶ **static channel model**
 - ▶ solution to share resources
 - ▶ abstraction of a transfer link
 - ▶ reflects hierarchical tier model
 - ▶ static configuration
 - ▶ configure SE **pairs**
 - ▶ mutual agreement between SE admins
 - ▶ network load is not taken into account
 - ▶ channel groups, star channel
 - ▶ still not optimal
- ▶ **transfer and control protocols**
 - ▶ bound to SRM + gridftp only

FTS 2 now

- ▶ **difficult FTS configuration**
 - ▶ YAIM, then config in XML files, in database, etc.
 - ▶ no usable default configuration
 - ▶ no consistent way of configuring
 - ▶ support burden
- ▶ **bound to Oracle**
 - ▶ limits FTS deployment (license)
 - ▶ job queue in database
- ▶ **redundant technologies**
 - ▶ Java web service
 - ▶ C++ agents
 - ▶ SRM web service client in C++
 - ▶ Oracle access in Java and C++
- ▶ **need explicit FTS endpoint to submit**
 - ▶ cannot select optimal FTS based on transfer parameters



FTS 2 sturcture



LCG_Util

- ▶ Data replication tool
- ▶ Multiple transfer protocols, SRM
- ▶ Register in file catalog
- ▶ Local <-> remote node as well
- ▶ Python, C API, CLI
- ▶ Client side tool

FTS and LCG_Util comparison

FTS

- ▶ asynch transfer execution
- ▶ third party transfers
- ▶ resource optimization
- ▶ gridftp only
- ▶ no LFC support

LCG_Util

- ▶ synch transfer execution
- ▶ local copies as well
- ▶ no optimization
- ▶ gridftp, rfio, dcap, file
- ▶ LFC support



Why new FTS?

- ▶ Architectural limits reached
- ▶ Difficult to maintain and improve
- ▶ Based on static channel model
- ▶ Uses Oracle backend only
- ▶ Overlaps with other projects
- ▶ Requires complex configuration
- ▶ Could handle resources better



FTS 2 and LCG_Util Schedule

- ▶ **2.2.6:** EMI. *June 2011.*
 - ▶ gridftp copy resume
- ▶ **2.2.7:** gLite. *July 2011.*
 - ▶ 2.2.6 for gLite
 - ▶ overwrite logic change
 - ▶ fix in gridftp checksum
- ▶ **2.2.8:** EMI, gLite. *Sep 2011.*
 - ▶ Glue2 support
 - ▶ monitoring messages
- ▶ **2.2.9:** EMI, gLite. *April 2012.*
 - ▶ optional
 - ▶ Oracle 11g integration
 - ▶ SL6 support
- ▶ **1.11.19:** gLite. *July 2011.*
 - ▶ FTS + GFAL common libs
 - ▶ gridftp copy resume
- ▶ **1.11.20:** EMI, gLite. *Aug 2011.*
 - ▶ Glue2 support
- ▶ **1.11.21:** EMI, gLite. *Sep 2011.*
 - ▶ Refactoring for FTS 3
 - ▶ trivial fixes

Roadmap:

<http://bit.ly/n9dp3g>

End of FTS 2. Roadmap:

<http://bit.ly/oVbVRh>



Main FTS3 milestones

Milestones: May be tested in production environment.

- ▶ **Milestone 1:** April, 2012
 - ▶ No channels
 - ▶ More transfer protocols
 - ▶ Performance comparable to FTS 2
- ▶ **Milestone 2:** December, 2012
 - ▶ Message based job queue
 - ▶ Dynamic resource optimization
 - ▶ Cover T2 use cases
 - ▶ LHC shutdown
- ▶ **FTS 3.0.0:** April, 2013.
 - ▶ Tests
 - ▶ Further optimizations

Roadmap: <https://svnweb.cern.ch/trac/fts3/roadmap>



Milestone 1

Remove channel model

- ▶ completely new FTS architecture
- ▶ configure SE-s only
 - ▶ separate SE from network
 - ▶ with remote config tool
 - ▶ SE-s may be queried for config
 - ▶ SE-s need to make it available (by M2)
 - ▶ global vs SE specific interface?
 - ▶ global: change in all SE-s, well...
 - ▶ SE specific: FTS plugins
- ▶ admin can select:
 - ▶ submit SE config for each FTS
 - ▶ publish SE config, FTS queries
 - ▶ survive with defaults



Milestone 1

Transfer protocols

- ▶ FTS uses GFAL plugins
- ▶ HTTP, xrdcp?

Configuration

- ▶ remote configuration tool (CLI)
- ▶ get/set config parameters in runtime
- ▶ configure FTS instances
- ▶ uniform way of configuration
- ▶ all configuration stored in database

Database plugins

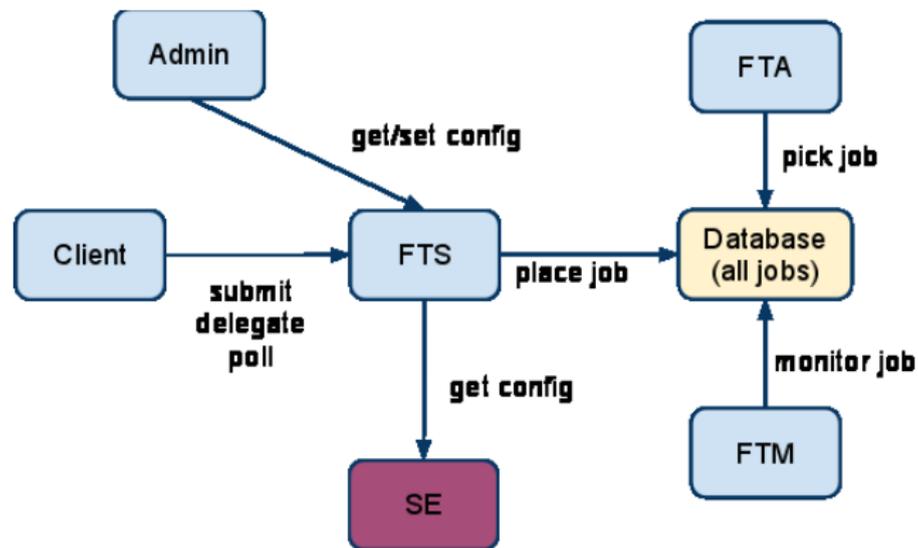
- ▶ Oracle only
- ▶ other databases can be supported in the future

Less technologies

- ▶ remove Java
- ▶ everything in C/C++



FTS 3 M1 structure



Milestone 2

Take network state into account

- ▶ How?
 - ▶ collaboration with OSG, with DYNES?
- ▶ learn from past transfers
- ▶ separate if network or SE is overloaded
- ▶ do not transfer on saturated link

Messaging based job submission

- ▶ submit to a global MQ
- ▶ best/authorized FTS executes the transfer: **pull** model
- ▶ load balanced/redundant FTS instances
- ▶ delegation problem
 - ▶ still need to contact FTS directly - an implicit step
- ▶ push model is still supported (direct job submission to an FTS)

Messaging based monitoring

- ▶ already in FTS 2.2.8

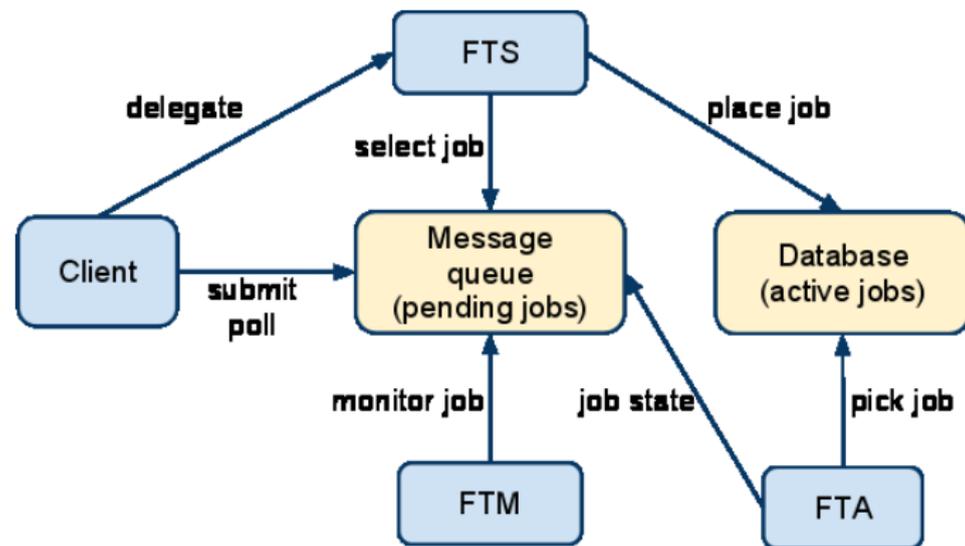


FTS 3 highlights

- ▶ works for Tier2-s as well
- ▶ more autonomous
- ▶ less transfer failures
- ▶ requires less configuration
- ▶ simpler to maintain and deploy
- ▶ uses standard building blocks
- ▶ uses LCG_Util components
- ▶ in production in 2013
- ▶ but provides values already in one year



FTS 3 sturcture



FTS and LCG_Util comparison

FTS

- ▶ asynch transfer execution
- ▶ third party transfers
- ▶ resource optimization
- ▶ **gridftp, HTTP, etc.**
- ▶ no LFC support

LCG_Util

- ▶ synch transfer execution
- ▶ local copies as well
- ▶ no optimization
- ▶ gridftp, HTTP, etc.
- ▶ LFC support

...but share a lot of code internally



Beyond FTS 3.0.0: Merge FTS and LCG_Util

FTS

- ▶ asynch transfer execution
- ▶ third party transfers
- ▶ resource optimization
- ▶ gridftp, HTTP, etc.
- ▶ LFC support (placement service?)

LCG_Util

- ▶ synch and asynch transfer execution
- ▶ local copies as well
- ▶ resource optimization
- ▶ gridftp, HTTP, etc.
- ▶ LFC support



Summary

- ▶ The FTS 2 development is over this year
- ▶ FTS 3 in production in 2013
- ▶ Better resource optimization: network, SE, code.



Thank you

<https://svnweb.cern.ch/trac/glitefts>

Jean-Philippe Baud

Oliver Keeble

Zsolt Molnar

Michail Salichos

+ our students and short term visitors

