

Table of Contents

Main feaues included in releases FTS 2.2 - 2.2.3.....	1
Changes from FTS version 2.1 to 2.2.3.....	1
SRM-gridftp split.....	1
Timeouts based on gridftp transfer markers.....	2
New transfer parameters.....	3
Delegation race condition.....	4
Logging improvements.....	4
Updated CLI.....	4
Site Groups (formerly known as clouds).....	4
Checksum support.....	5
Fixed bugs.....	5
Known Issues.....	5

Main features included in releases FTS 2.2 - 2.2.3

Changes from FTS version 2.1 to 2.2.3

SRM-gridftp split

Channel occupation is now calculated differently for url-copy channels.

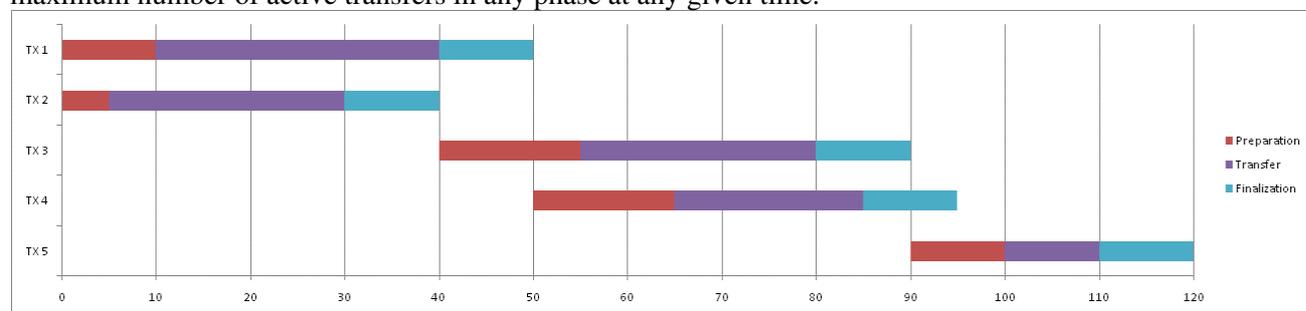
FTS transfers in url-copy channels are divided in three phases: preparation, transfer and finalization.

During the preparation phase FTS contacts the two SRM endpoints to make sure that all the prerequisites for the transfer to be successful are met: the servers are reachable, the source file exists, the target directory exists and so on. During the finalization phase, FTS communicates to the servers that the transfer has completed successfully, the TURLs can be released and so on. It is only during the transfer phase that data transfer is actually taking place and bandwidth is used.

To illustrate the different scheduling mechanism, let's assume that some transfers are scheduled on a channel configured with 2 files, where the duration of the different phases for each transfer is summarized in the table below.

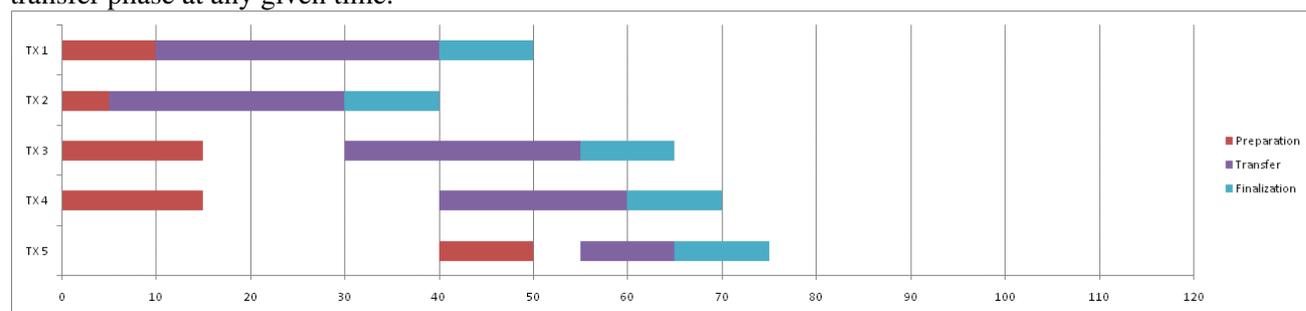
TX ID	Preparation phase	Transfer phase	Finalization phase
TX 1	10	30	10
TX 2	5	25	10
TX 3	15	25	10
TX 4	15	20	10
TX 5	10	10	10

With FTS versions up to 2.1, the limit on the number of files configured on a channel was meant as the maximum number of active transfers in any phase at any given time.



This scheduling mechanism is not very efficient. See, for example, that Transfer 3 start only when transfer 2 is completed: in the interval from 40 to 50 seconds there are 2 active transfers, but none of them is actually transferring data.

With FTS version 2.2, the number of files on a channel is meant as the maximum number of transfers in the transfer phase at any given time.



In this example, at the beginning, 4 transfers instead of 2 are started (this ratio is configurable, see **INSERT LINK** for details). Transfers 1 and 2, the first reaching the end of the preparation phase, switch immediately to the actual transfer phase, and as soon as they end their transfer phases other ones are started so that there

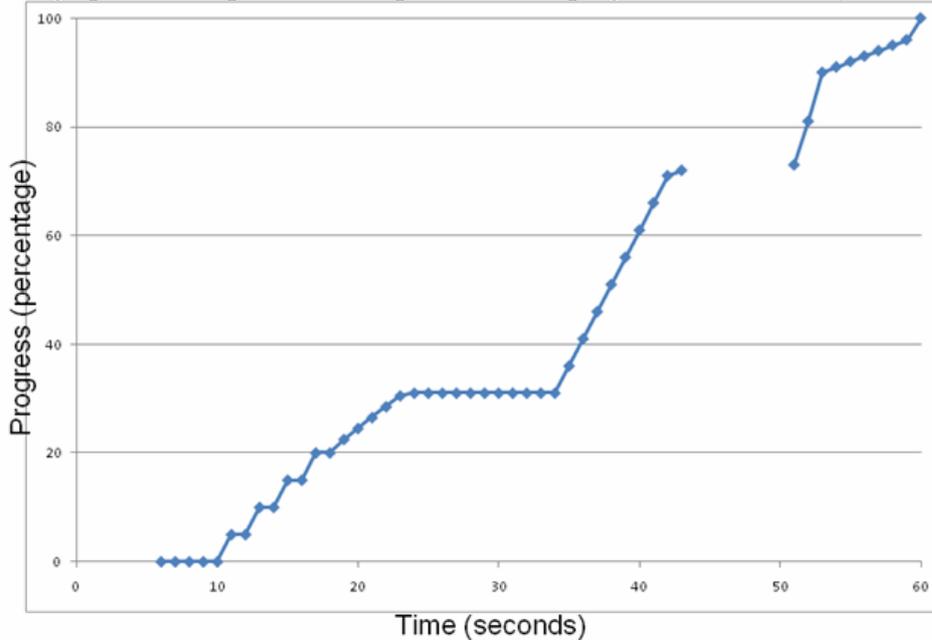
are always 2 actual data transfer processes.

From FTS 2.2.3, you can choose between the old and new behavior. See the GUC_SRMGRIDFTPSPLIT Yaim variable in the FTS Configuration Reference [page](#). By default, the new behavior is **disabled!**

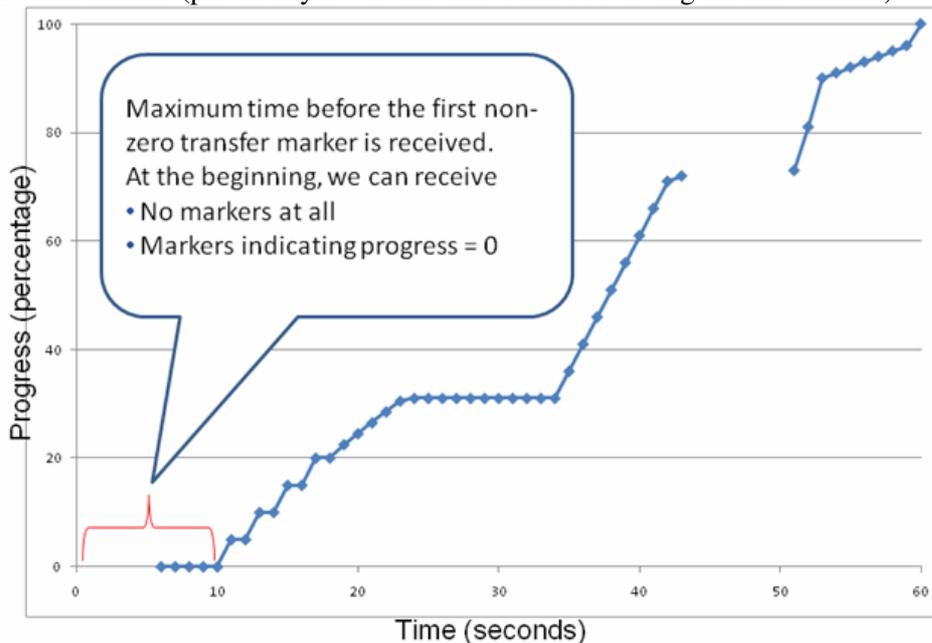
Timeouts based on gridftp transfer markers

The timeout parameters based on gridftp transfer markers have been reviewed. Transfer markers indicate that the server is alive and give information about the transfer progress.

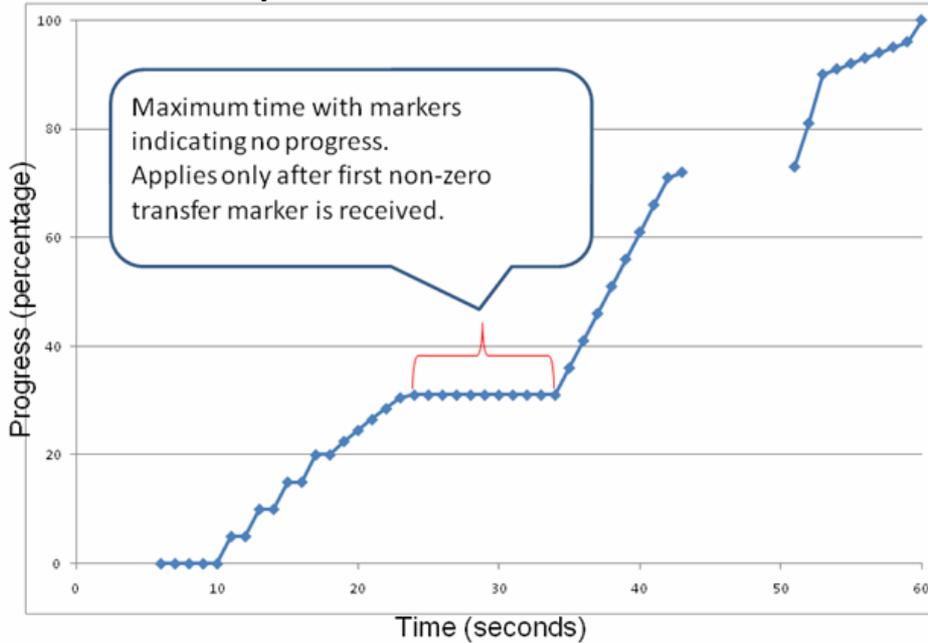
This graph shows a possible example of transfer progress as indicated by the received markers:



The meaning of **first transfer marker timeout** has been slightly modified from version 2.1: it is now considered as the maximum amount of time, after the transfer starts, before receiving the first non-zero transfer marker (previously it was the time before receiving the first marker):

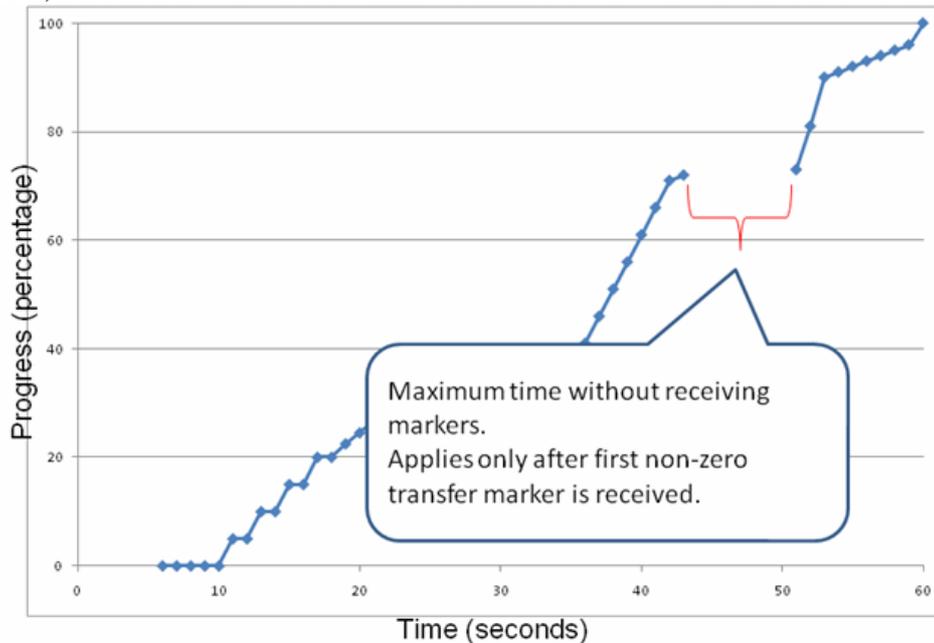


The newly introduced **no activity timeout** is the maximum amount of time with markers showing no



progress:

As in previous versions, the **transfer markers timeout** indicates the maximum amount of time without



receiving markers:

New transfer parameters

Handling WLCG requests for extra transfer parameters. For the "Addendum to the SRM v2.2 WLCG Usage Agreement" document check the CCRC'08 Wiki Page.

Copy pin lifetime: a copy pin lifetime can be specified with the `--copy-pin-lifetime` of the `glite-transfer-submit` command line tool.

For urlcopy channels, copy pin lifetime is set in two places:

- in the storageSystemInfo array of key-value pairs of the srmPrepareToPut operation, in a key called CopyPinLifetime

- in an additional `srmBringOnline` operation performed after the copy has succeeded (successful `srmPutDone`), in the `desiredLifetime` parameter.

For `srmcopy` channels, copy pin lifetime is set in two places:

- in the `targetStorageSystemInfo` array of key-value pairs of the `srmCopy` operation, in a key called `CopyPinLifetime`
- in an additional `srmBringOnline` operation performed after the copy has succeeded, in the `desiredLifetime` parameter.

The additional call to `srmBringOnline` is a workaround to deal with storage elements that will not provide a short term implementation of the `ExtraInfo` parameters handling, and will probably be removed in next versions.

Source space token in `srmCopy` operations: Now the source space token option is set for `srmcopy` channels as well, in the `sourceStorageSystemInfo` array of key-value pairs of the `srmCopy` operation, in a key called `SourceSpaceToken`. The source space token was already correctly handled in `srmPrepareToGet` operations for `urlcopy` channels.

Transfer type: it is now possible to specify `ConnectionType=LAN` in the `TransferParameters` field of the `srmPrepareToGet` / `srmPrepareToPut` operations. Use the `--lan-connection` flag of `glite-transfer-submit`. This should be useful for intra-site transfers involving `dCache` storage elements.

Fail nearline files: fail the transfer if the source file location is nearline, without attempting an `srmPrepareToGet` operation. Specify it with the `--fail-nearline` flag of `glite-transfer-submit`.

Delegation race condition

Fix for bug #60095: FTS: Couldn't set the private key `glite-data-transfer-fts v3.7.0-3 2010-01-18`

Logging improvements

Logging to `syslog` according to `Middleware Security Audit Logging Guidelines`.

Logging IP addresses in SRM calls.

Updated CLI

Site Groups (formerly known as clouds)

The CLI now correctly handles all the **channel configuration parameters** that have been moved to the database.

Added **site groups management tools**: `glite-transfer-group-list`, `glite-transfer-group-addmember` and `glite-transfer-group-removemember`.

The `glite-transfer-submit-placement` and `glite-transfer-discovery` commands are now **obsolete**, and no longer distributed.

See the documentation for `FTS 2.2 CLI` [for details](#).

Checksum support

A client can specify checksums and ask FTS to check it with the source and/or destination SE via srmLs or gridftp calls:

```
glite-transfer-submit --compare-checksums source-SURL destination-SURL ADLER32:12345678
```

See more at FtsChecksums.

Fixed bugs

For a list of fixed bugs, see patches

FTS version	PATCH SLC4/32
2.2	SLC4 i386 ↗
2.2.1	SLC4 i386 ↗
2.2.2	SLC4 i386 ↗
2.2.3	SLC4 i386 ↗

For upcoming releases see the FTS patch status page!

Known Issues

See Fts223KnownIssues

Last edit: UnknownUser on 2010-05-11 - 18:38

Number of topics: 1

Maintainer: AkosFrohner

This topic: EGEE > FtsRelease22

Topic revision: r10 - 2010-05-11 - unknown



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

Ideas, requests, problems regarding TWiki? Ask a support question or Send feedback