File Transfer Service Monitoring System for LCG

A.V. Uzhinskiy

Laboratory of Information Technologies, JINR

Аннотация

Рассматривается концепция системы мониторинга сервиса передачи данных (FTS) пакета glite в LHC computing grid. Система предназначена для обнаружения ошибок и нестандартных ситуаций на каналах передачи данных. Предложена трехуровневая архитектура системы - сбор данных, хранение и размещение данных, представление данных. Первый уровень реализован на языках Perl и Shell, второй - с использование СУБД MySql, третий - PHP и XHTML. Представлено краткое описание интерфейса пользователя и отчетные формы - количество ошибок на каналах, различные статистические отчеты и т.д. Работа проводится в рамках сотрудничества ОИЯИ, НИИЯФ МГУ и ЦЕРН.

The most important way of Grid services evolution now-a-days is an improvement of their performance and reliability: users want to have stable access to resources 24 hours a day, 7 days a week, in spite of holidays and vacations. A good monitoring system can assist here by providing a global view of the system, which can be helpful in identifying performance problems and in detecting mistakes, non-standard situations and bottlenecks of the system.

Until very recently, the set of monitoring tools for FTS (File Transfer Service) provided only several possibilities: an analysis of data transfer between channels using GridView [1] and a representation (FTS Report [2]) of statistics concerning all the transfers performed for last day or last week with percentage of failure and two most popular failure reasons. There are also several tools developed at Tier-1 sites which provide information about a current system situation (status of agents, channel settings and etc.) throw a web-interface without necessity to use a command line.

There were a lot of uncovered questions about the system functionality and unacceptable delays in receiving the channel problem information. It is a reason of developing the monitoring system "Spider", the main aim of which is a providing of detailed information about errors appearing on the FTS-channels. The work is conducted within the LCG project in cooperation with JINR, MSU and CERN.

FTS is arranged in such a way that all information about transfers is stored as log-files on servers where transfer-agents are installed. Directories with the files have been cleared by a cron job once a day at midnight. One can get a more comprehensive information about the structure and the architecture of the FTS at [3, 4]. We have choosen a tree-layer architecture for our monitoring system. The first layer is a data mining, a set of scripts for transfer and processing of the log-files (Perl + shell). The second layer is a data storing, a storing of all information taken from the log-files in a convenient way (MySql). The third one is a data representation which provides the user with stored data using a flexible interface (PHP + XHTML).

Data mining (Perl + shell): First step is a deliver of the log-files from servers where the transfer agents work to the point where the monitoring system is installed. In principle, we could manage without the step because log processing could take place directly on the servers, but to decrease loading on the major units of the FTS, we decided to execute all the works on independent machines. Scripts run every hour. Such interval

allows administrator to react on a trouble situation in channels in a proper time. The run does not arise excessive system loading.

A next step is a gathering of a statistical information about various errors appeared on the channels from midnight. It should be mentioned that FTS interacts with a lot of data storing and data transfer applications such as dCach, Castor, dpm, gridftp, etc. So, the amount of mistakes leading to the transfer failure is pretty large, and we try to provide an administrator with a simple mechanism for adding in the monitoring system a new definition of mistakes. Massages about mistakes usually contain a lot of useless information like "DESTINATION error during PREPARATION phase: [FILE_EXISTS] at Tue Jul 24 13:18:28 CEST 2007 state Failed : file exists". Thus we use a mechanism of patterns for an easily manage with errors. The pattern is a unique part of the error massage certainly referring it to a concrete class. To store errors in our system, we use records consisting of full error descriptions, three error patterns and other technical information.

A simplified algorithm of our log processing script looks as follows: Running throw the all known mistake patterns in the date base and searching for all error patterns in the text of current log file. If there is a full conformity, a mistake counter of the defined class of channel is increased by one. If at the end of the cycle the error has not been identified, it should be marked as unknown and sent to an administrator. We have multiplicities of the errors of every types on every channels at the end of the script work. After that we write the information in our data base.

Data storing (MySql): We use several tables to store information about main objects in the system: CHANNEL is a table consisting information about monitored channels; MISTAKE is a table with known errors containing samples, patterns, types etc; KMDB - historical table where all cases of the errors appearing on the channel are described with information about time and reason of an error; All information about multiplicities of different errors appearing on channels is stored in the table MAIN.

Data representation (PHP + XHTML): The WEB-interface of the monitoring system is located at https://fts102.cern.ch/spider/channel.php
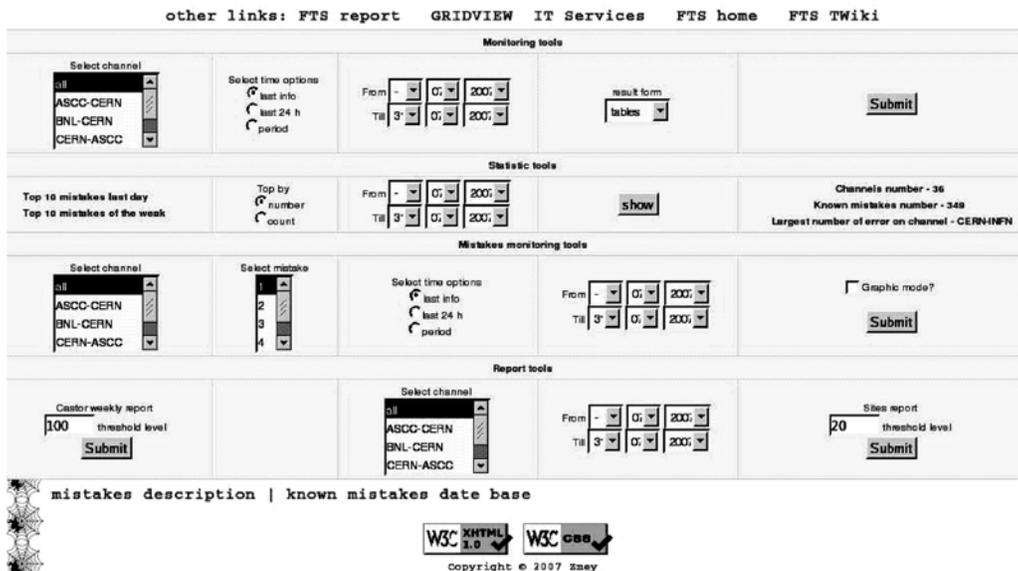

Fig. 1: Interface of the monitoring system

The main unit of the program "Monitoring tools" has several settings - channel, time

interval and result form. The user can get current error information, information for last 24 hours, or information for any time interval more then one day. Result form provides the user with tables, charts, graphics and also errors aggregated by type (analytic).
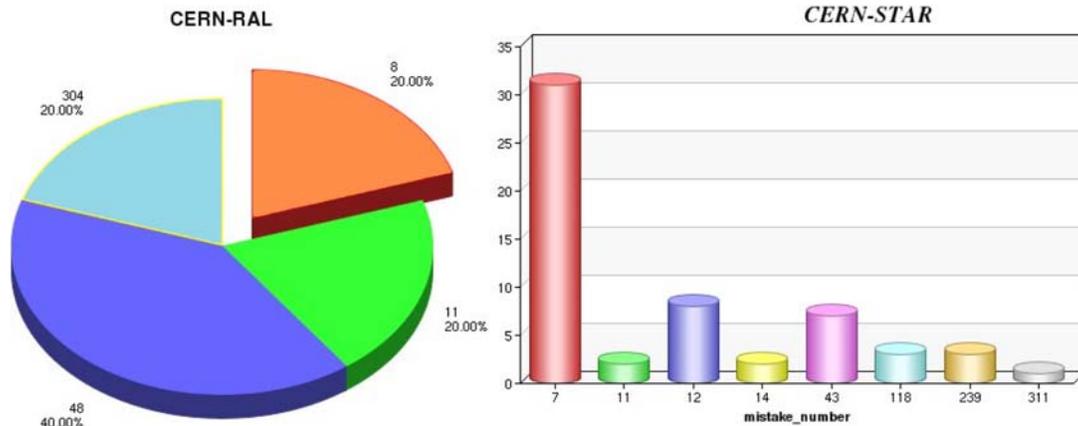

Fig. 2: Example of the system reports

Usually we can define which application raises an error - dCach, Castor, dpm. gridftp, etc. We can also say where the error has appeared - data source site, destination site or during transfer. We have analyzed all the errors and have clearefied 16 main error types: *fts* is an error caused by logical or program mistakes in FTS; *globus* is a gridftp error; *user* is an error like wrong file name, etc; *tcp* - network connection problems; *t_dcach, s_dcach, dcach* - dCach errors ("t_" and "s_" marks on which side the error has appeared "s" - source, "t" - target. "" - undefined side); *t_castor, s_castor, castor* - Castor errors; *t_dpm, s_dpm, dpm* - dpm errors; *t_srm, s_srm, srm* This class gathers not only srm errors but all other errors for which we could not define a certain class.

date	time	all	lost	fts	globus	user	tcp	dcach	castor	t_dcach	s_dcach	t_castor	s_castor	dpm	t_dpm	s_dpm	t_srm	s_srm	srm
2007-07-31	09:28:59	3985	21	0	12	0	0	0	2	840	0	1978	0	0	0	0	1151	0	2

Fig. 3: Analytic report about CERN-INFN channel

Channel	21	22	42	48	49	71	72	87	91	92	177	178	199	287	341	349
CERN-INFN	421	0	0	0	0	0	0	0	0	0	0	0	0	0		1557

Fig. 4: Same report opened by "t_castor" class

One can find some interesting links on other FTS-facilities at the our monitoring system page. There is also a list of all errors in the system - http://pcitgd26.cern.ch/ mistakes.php, and information about cases of error appearance on the channels with they reason - http://pcitgd26.cern.ch/mkdb.php.

If result type is *tables* or *analytic*, the user has to move cursor to the error id-number to get a text of the error (picture 5). To get a full text of the error with a detailed description, the user has to click on the error id-number (picture 6).

Date	Time	all	unrec	failed	lost	unknown	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2007-08-02	08:25:56	872	0	872	25	0	0	0	0	0	0	0	0	54																								0

the server sent an error response: 553 553 Address already in use.

Fig. 5: Example of pop up prompt

[314] TRANSFER during TRANSFER phase: [INVALID_SIZE]

2007-07-14[INFN]

Reason:/castor/cnaf.infn.it/grid/lcg/cms/tape/mc/2006/12/22/mc-physval-120-BBbar80to120-LowLumiPU/0008/2C343C8A-9CB9-DB11-8857-0030487219AB.root was corrupted, size of real file < of size in the name server.probably it is due to a filesystem problem occurred months ago.

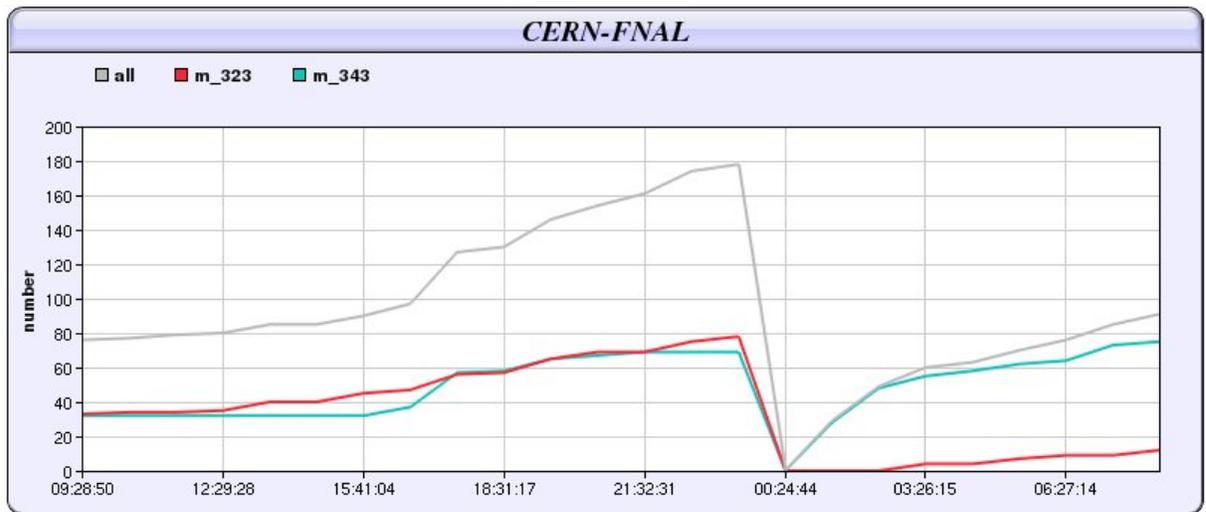Fig. 6: Page with detailed error description

Fig. 7: Report about mistakes 323 and 343 on CERN-INFN channel for last 24 hours

Place	Error_N	Number	Description
1	321	287005	DESTINATION during PREPARATION phase: [GENERAL_FAILURE] destination file failed on the SRM with error [SRM_FAILURE]
2	314	223045	TRANSFER during TRANSFER phase: [INVALID_SIZE]
3	12	204788	451 Local resource failure: malloc: Cannot allocate memory
4	313	202982	SOURCE during PREPARATION phase: [REQUEST_TIMEOUT] failed to prepare source file in 180 seconds
5	306	182382	failed to prepare Destination file in 180 seconds
6	332	158218	DESTINATION during FINALIZATION phase: [ ] been successfully aborted
7	8	119124	the server sent an error response: 425 425 Ca
8	349	90901	DESTINATION during PREPARATION phase:
9	55	90756	the server sent an error response: 425 425 Ca

Place	Error_N	Count	
1	11	1379	an end-of-file was reached
2	313	1335	SOURCE during PREPARATION phase:
3	311	1047	TRANSFER during TRANSFER phase: [T
4	12	885	451 Local resource failure: malloc: Cann
5	14	797	the server sent an error response: 426 42
6	8	786	the server sent an error response: 425 42
7	239	605	the server sent an error response: 553 55
8	306	551	failed to prepare Destination file in 180 s
9	21	468	DESTINATION during PREPARATION phase: [GENERAL_FAILURE] CastorStagerInterface.c:2507 Device or resource busy (errno=0, serrno=0)

Fig. 8: Last week rating and 27.07.2007 - 01.08.2007 most frequent errors

We have developed a module "Mistake monitoring tools" for cases when we are not interested in all mistakes, but are looking only for a few of them during a debugging process. It allows us to get information about a particular mistake in time. The module

has almost the same settings as a previous one. *Result type* can be only present as a table or a graphics. The users can choose certain mistakes. Information about them will be displayed.

Next module "Statistic tools" is created to provide main information about the system - number of channels and mistakes, channel with highest mistakes multiplicity, etc. The user can get errors sorted by frequency or total amounts on channels for last day, week or other time interval using this module.

"Reports tool" module is aimed for an ease creation of various reports. It produces a weekly report about problems appearing on CERN side. It also allows one to get a report about the most frequent errors on Tier-1 sites with a defined threshold level for any time interval.

CERN-INFN threshold level 20 (2007-7-21 - 2007-8-1)

all	average	max	min	m_id	m_example	total
5442	454	1038	0	21	DESTINATION during PREPARATION phase: [GENERAL_FAILURE] CastorStagerInterface.c:2507 Device or resource busy (errno=0, serrno=0)	66920
726	61	425	0	32	SOURCE during PREPARATION phase: [INVALID_PATH] specified file(s) does not exist	66920
1955	163	840	0	52	a system call failed (Connection refused)	66920
16128	1344	3580	0	321	DESTINATION during PREPARATION phase: [GENERAL_FAILURE] destination file failed on the SRM with error [SRM_FAILURE]	66920
22933	1911	4576	0	306	failed to prepare Destination file in 180 seconds	66920
10302	859	4047	0	349	DESTINATION during PREPARATION phase: [GENERAL_FAILURE] CastorStagerInterface.c:2507 Permission denied (errno=, serrno=)	66920

Fig. 9: CERN-INFN most frequent errors for period 21.07.2007 - 01.08.2007

The system is currently used for the maintenance of the FTS channels at CERN. We are working on its further development.

References

[1] http://gridview.cern.ch/GRIDVIEW/index.php

[2] https://fts102.cern.ch/ftsstats/prod-fts-ws

[3] http://egee-jra1-dm.web.cern.ch/egee-jra1-dm/FTS/default.htm

[4] https://twiki.cern.ch/twiki/bin/view/EGEE/FTS