No file left behind

Monitoring transfer latencies in PhEDEx

T. Chwalek[1], O. Gutsche[2], C.-H. Huang[2], R. Kaselis[3], M. Klute[4], **N. Magini[5]**, F. Moscato[2], S. Piperov[6], **N.Ratnikova[1,7]**, P. Rossman[2], A. Sanchez-Hernandez[8], A. Sartirana[9], T. Wildish[10], S. Xie[4], M. Yang[4]

[1]Karlsruhe Institute of Technology, [2]Fermi National Accelerator Laboratory, [3]Vilnius University, [4]Massachusettes Institute of Technology, [5]CERN, [6]INRNE, Bulgarian Academy of Sciences, [7]ITEP, [8]Centro Invest. Estudios Avanz. IPN, [9]École Polytechnique, [10]Princeton University

The CMS experiment at the LHC uses **PhEDEx** to distribute data among the sites of the Worldwide LHC Computing Grid (WLCG), transferring over 500 TB per week since the beginning of LHC data taking.

To avoid data loss and unavailability, PhEDEx is designed for 100% completion of transfer subscriptions even on an unreliable infrastructure. This is achieved with small impact on global performance and low operational cost with an intelligent automation of the retry of failed transfers.

However, a large amount of operator effort is still needed to identify transfers that are permanently stuck and need manual intervention to reach full completion. For this reason we have decided to instrument PhEDEx with a **latency monitoring system** that can be used to alert operators.

Block latency monitoring in PhEDEx 4.0

The minimum unit for PhEDEx subscriptions is the file block.
To achieve scalability, PhEDEx tracks the states of individual files only during transfers, cleaning up the database regularly.
Until PhEDEx 4.0, transfer latency records were only available at the level of data blocks.

CMS data management structures

- File ~ 1 GB for efficient handling on storage and worker nodes
- File block ~ 1 TB to reduce the complexity of data management
- Dataset – up to 100 TB by physics content

File latency monitoring in PhEDEx 4.1

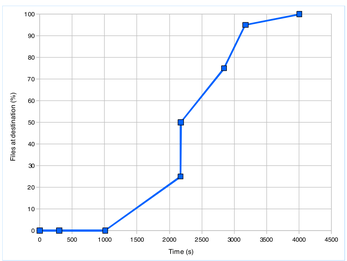
The design for the latency monitoring system was finalized in a codefest in August 2011. To collect file-level latency details without affecting transfer management performance, we used separate tables for live data and for historical logs. The scalability of the new schema was demonstrated on a Testbed in late 2011 and early 2012, running a simulation of PhEDEx transfers at 10X-100X the production scale.

Latency monitoring tables

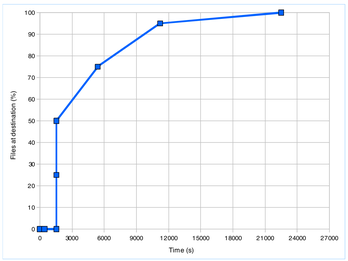
- **t_dps_block_latency**
  - Events for blocks currently in transfer: subscription, suspension, completion, etc.
- **t_xfer_file_latency**
  - Events for files in incomplete blocks: transfer routing, attempt, success, etc.
- **t_log_block_latency**
  - Archive t_dps_block_latency indefinitely, aggregating file info from t_xfer_file_latency
- **t_log_file_latency**
  - Archive t_xfer_file_latency for 30 days

Examples of latency measurements

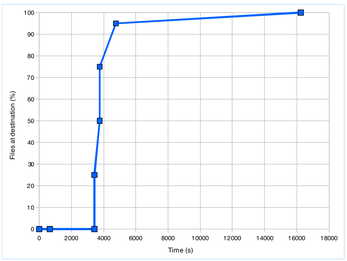
The Testbed simulation demonstrated the possibility to recognize different patterns in block completion latencies.



Block with perfect transfer quality: close to linear (some irregularity due to agent cycle times)



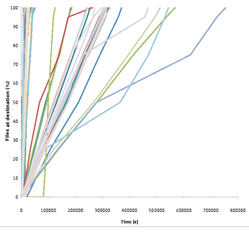
Block transferred on a link with high error rate (20%) completed after multiple retries



Block with perfect transfer quality except for 10 stuck files requiring manual intervention (“transfer tail”)

Status and outlook

PhEDEx 4.1 is already collecting transfer latency metrics for LoadTest file transfers.



PhEDEx 4.1 will soon start to collect latency metrics for production file transfers, to be used to plan the data placement strategy and identify areas where development effort is needed.
We are also developing a web interface for the latency monitoring system to provide plots and alerts for the operators.