Monitoring data transfer latency in CMS computing operations

[1]Andrea Sartirana, [2]Daniele Bonacorsi, [3]Meric Taze, [4]Nicolo Magini, [2]Tommaso Diotalevi, [5]Tony Wildish

[1]Ecole Polytechnique [2]University of Bologna [3]Cukurova University [4]Fermi National Accelerator Laboratory [5]Princeton University

The CMS experiment at the LHC has developed and is running the PhEDEx system to manage data transfers over WLCG sites. PhEDEx has transferred a total of 150 PB during Run-1, and it is currently moving about 2.5 PB per week among ~60 sites. The PhEDEx design aims at providing the highest possible transfer completion rate, despite possible infrastructural unreliabilities, achieved via intelligent fail-over tactics and automatic retries. During the 11 years of its operations, including the first LHC data taking period (Run-1) and the first LHC Long Shutdown (LS1), a large set of data concerning the latencies observed in all transfers between all Tiers has been collected. The study of this data set allows a categorization of the different root causes of such latencies, helping to shape the strategy to attack this problem and increase the overall performance of the PhEDEx system

Cleaning the data & defining variables

In the last 2 years PhEDEx collected transfers latency data for roughly 3M blocks. They have been skimmed removing:

* ill defined entries and test blocks

* transfers lasting less than 3h (no latency issues)

* suspended transfers or transfers of open blocks (complex treatment)

We are left with roughly 120k entries on which our analysis was performed. Moreover, tails analysis requires defined values for the "skew" variables as well as data blocks which are big enough to define a "bulk" and a "tail". Thus, data for this analysis was further skimmed keeping only the 45k blocks with size larger than 300GB and with defined values for all the "skew" variables.

Skew variables show transfer rate ratio between the time spent in a small portion(last 5% or first 25%) and X percent from the beginning/end of a transfer.

-**skew X**: (time spent transferring the LAST 5 percent of the files) / (time spent transferring the FIRST X percent of the files) times X/5
-**skew last X**: (time spent transferring the LAST 5 percent of the files) / (time spent transferring the LAST X percent of the files) times X/5
-**reverse skew last X**: (time spent transferring the FIRST 25 percent of the files) / (time spent transferring the LAST X percent of the files) times X/25

Latency of blocks that get stuck early

Although CMS tries hard to detect the problematic sites in advance and proactively takes measures to use them only if they are safe both for processing and for data transfers, it is not always possible to select with 100% purity on long periods of time a subset consisting of only sites in perfect shape. Hence, it is expected that the overall transfer system needs to always deal with a bunch of problematic sites that should nevertheless be used as either source or destination of some data transfer tasks. In many of these cases, as the problem may just be at the infrastructural site level, these kind of transfers show up as “stuck” in the very beginning, i.e. even in the transfer of the very first file.

These transfers are hence reported as stuck at 0% completion for hours as shown in the Figure 4, so it is far easier to detect them with respect to any other latency type.

In addition, skew last 75 values (rate in last 5% : rate in last 75%) are quite small while skew 25 (rate in last 5% : rate in first 25%) have quite large values in the Figure 3 as expected.

The solution, however, is not straight-forward: it might require site admin intervention at the source/destination site, or even central operators/experts involvement. The price to pay if not promptly identified is high: only a quick problem identification, attack and fix can avoid to pile up delays and additional work load at a later stage

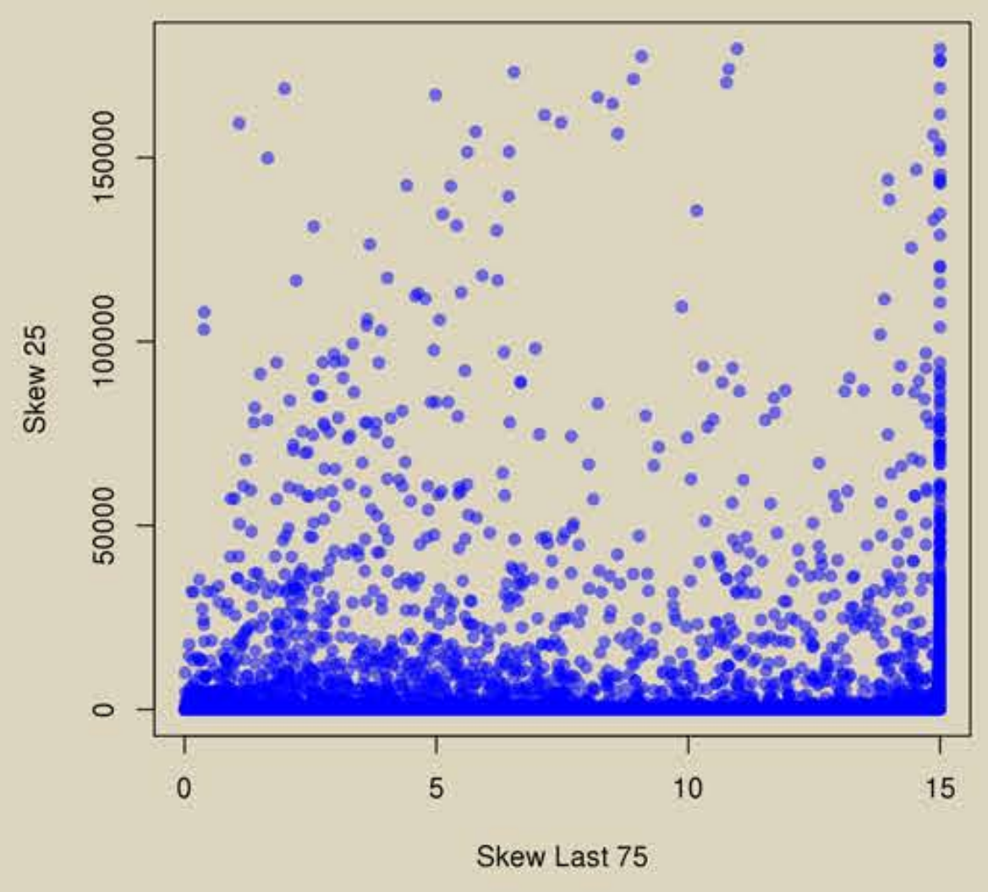

Figure 3

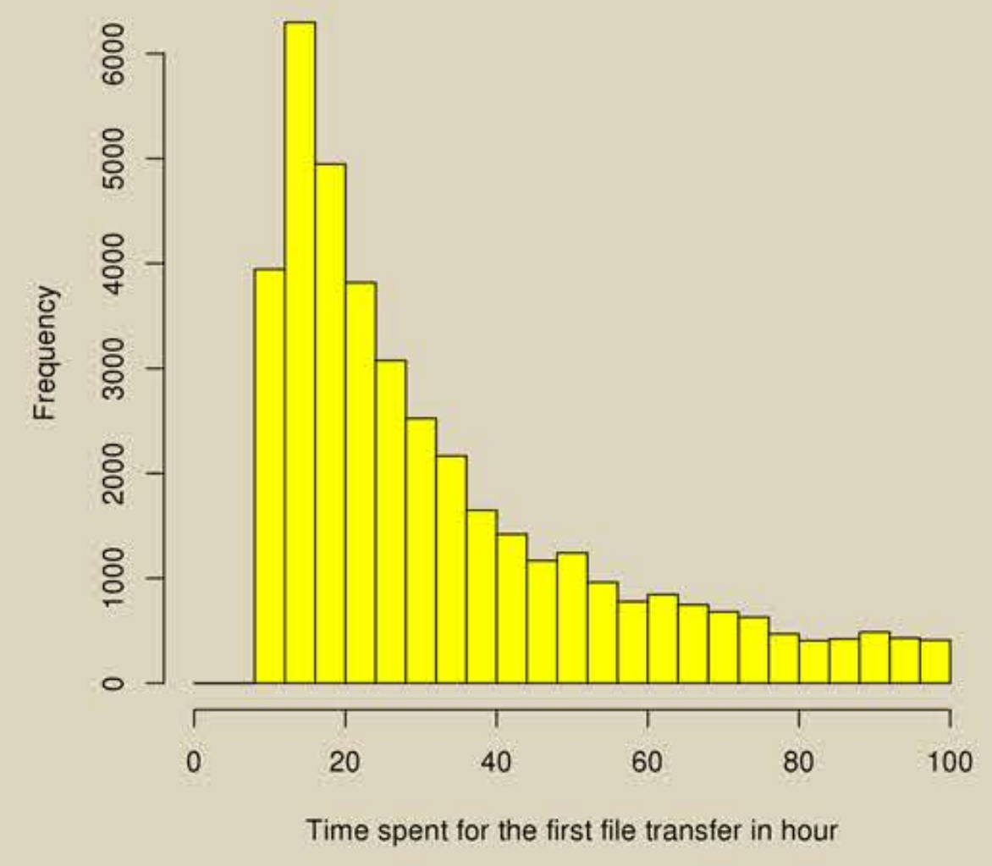

Figure 4

Instrumenting PhEDEx with detailed transfer latency monitoring

The atomic unit for transfer operations in PhEDEx is the **file block**: an arbitrary group of O(100-1000) files in the same dataset. To achieve scalability, PhEDEx doesn't keep a permanent record of the states of individual files: all information is aggregated at the level of block after transfers are completed. This level of detail is sufficient for replica location, but it is not enough to identify problems that increase latency in block transfers.

For this reason, in 2012 we instrumented PhEDEx with a detailed latency monitoring system, collecting file-level information on transfers in historical monitoring tables separated from the live transfer tables to avoid affecting the transfer performance. Events related to transfers of individual files (time of first transfer attempt, number of retries, time of final successful transfer attempt) are recorded and archived for 30 days. These events are aggregated at block level and archived indefinitely to keep a history of the most significant events in the completion of each block transfer: time of the block subscription, time of the first successful file transfer, time of the first 25%/50%/75%/95% file transfers, time of block completion.

Latency of big blocks with long tails

Due to operational problems at the production level, it may happen that a few files are not produced properly. They might either be missing (i.e. not even written to storage) or being corrupt with a wrong size/checksum. With the help of PhEDEx pre/post validation scripts as well as FTS internal mechanisms, the transfer of these files is reported as a failure back to PhEDEx. In order to achieve the best throughput in transfers, PhEDEx puts such failing files back into the transfer queue while transferring other files, and suspends these transfers for a longer time when all transfer attempts for a file fail continuously.

The impact of these missing/corrupt files can only be seen in the last few percentages of the transfer of the whole block. Hence, while transfer rates in the first 95% have higher values, rate in the last 5% drops off to quite low values as shown in the Figure 2. In a similar way, reverse skew last 5 values (rate in first 25% : rate in last 5%) are much lower than skew 95 values (rate in last 5% : rate in first 95%) in Figure 1.

Solving this problem requires a manual expert operator intervention, consisting of either replacing the file (if it has other valid replicas) or otherwise invalidating it and announcing it as lost.

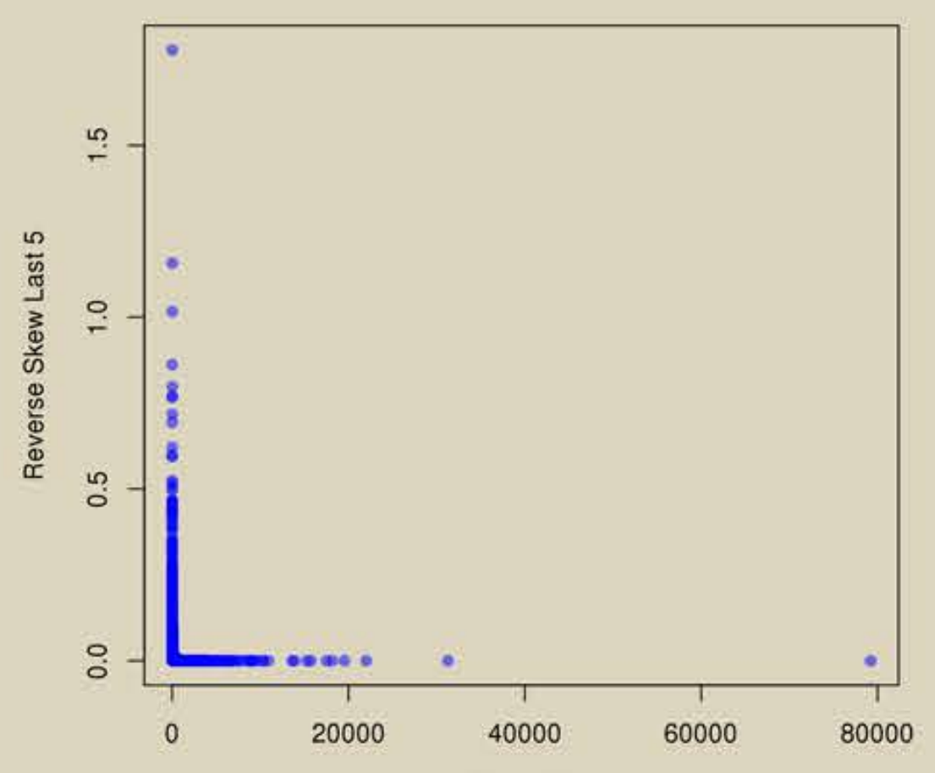

Figure 1

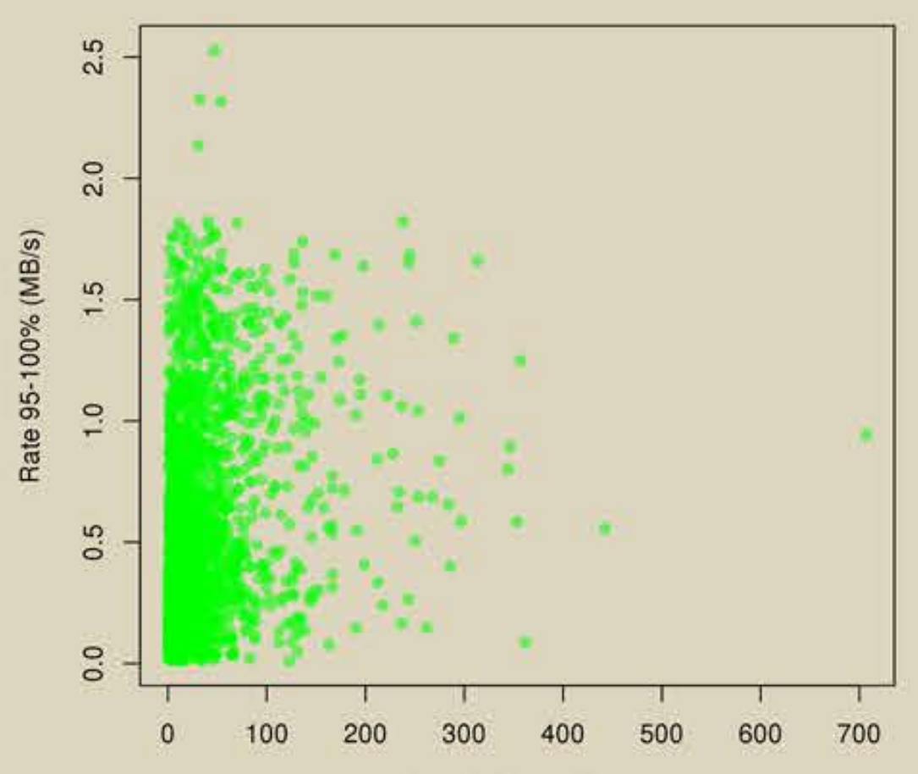

Figure 2

Latency of datasets with many small blocks

In all CMS processing activities, the growth of a CMS dataset to be intended as a collection of files with specific physics con tent happens through the contribution of production efforts by different WLCG sites supporting the CMS workflows, and in many sites contributing to a dataset are not the final destination where such dataset is supposed to be stored. Hence, a PhEDEx subscription is made to a tape endpoint and possibly to some disk endpoints to gather all produced data into the needed places according to CMS policies. However, some of the blocks might be located at a problematic site that can cause latency in dataset level transfers.

The underlying problem can be missing/corrupt files, or storage related problems as mentioned in other latency types.

CHEP2015: 21st International Conference on Computing in High Energy and Nuclear Physics
13-17 Apr 2015, Okinawa Institute of Science and Technology, Okinawa (Japan)