2018 IEEE 14th International Conference on e-Science

Modelling High-Energy Physics Data Transfers

Joaquin Bogado, Fernando Monticelli, Javier Diaz
Universidad Nacional de La Plata, Argentina
Joaquin.Bogado@cern.ch

Mario Lassnig
CERN, Switzerland
Mario.Lassnig@cern.ch

Ilija Vukotic
University of Chicago, Illinois
ivukotic@uchicago.edu

EXTENDED ABSTRACT

In scientific data management systems like Rucio[1], the possibility to know when a file transfer is going to be finished at the moment of submission opens a wide range of opportunities to improve the schedule techniques actually being used, and therefore to optimize the use of the available resources.

We developed a model that can predict the number of pending transfers in a file transfer system[2] queue at a given time, and therefore, with some level of confidence, the estimated time to complete for each transfer. Using data analytics methods on historical data, we also managed to make predictions about the average rate of the transfers based only in their sizes.

The models use information about the submission time stamp, i.e., the moment the transfer enters to the data management system, and the size of the transfer in bytes, to calculate the starting time stamp, i.e., the beginning of the usage of the network, and finishing time stamp. The rate of each transfer needs to be known or approximated. Also, the limits of concurrent active transfers need to be known. We got the rate approximation doing fit using ordinary least squares regression from scipy optimize package[4] to the function described in Equation (1) on 500 random transfers in the first dataset.

$$T_{rate} = T_{size}/((T_{size}/rate) + overhead) < disklimit \quad (1)$$

Here, $rate$, $overhead$ and $disklimit$ are the parameters of the fit. T_{rate} is the average rate for a transfer of a file of T_{size} bytes. The $overhead$ parameter is associated to TCP connection delays, which we assume exist for every transfer. Also, $disklimit$ is a parameter that describes potential throughput limitations or delays of the involved storage systems.

The observed average rate and maximum number of active concurrent transfers can be calculated. From this data, it is possible to know a posteriori how much time a particular transfer spent in the queue and how much time the actual transfer took on the network. Models vary slightly in the way they calculate the available bandwidth for each transfer. The observed average rate for each transfer cannot be calculated before the transfer ends. Two of the models use a posteriori analysis from historical data to calculate the observed rate of each transfer. The three other models use a method to approximate the observed average rate for each transfer relying only on information available at submission time. These models differ slightly in the way they limit the number of active concurrent transfers in the link during the simulation.

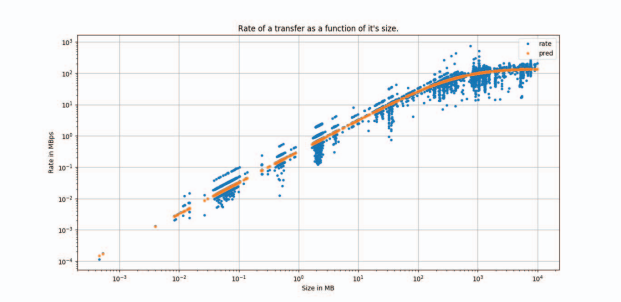

Fig. 1. Average rate of a transfer as a function of its size for transfers in the first dataset, in blue. Prediction for the average rate from a fit of Equation 1 over 500 random observed points, in orange.

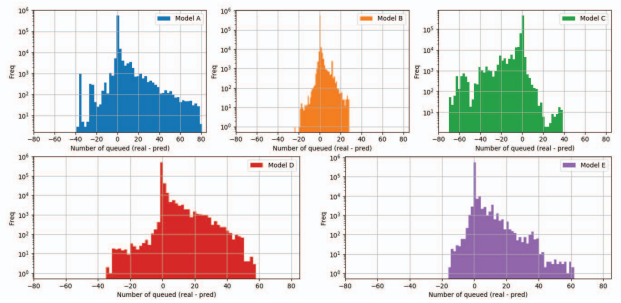

Fig. 2. Histogram of the errors (observed - predicted) in the number of queued files for 600K seconds of simulation of the first dataset.

The models were tuned using a sample dataset that represents the transfers of one week. Validation was done using a second dataset with transfers of a different week.

Table I shows the errors that allow to compare the models against each other. The Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R^2 score, average of $(observed - prediction)$ or (μE), and standard deviation of $(observed - prediction)$ or (σE) is calculated. Models B, D and E show comparable results in both the first and second dataset, with zero-centered μE and small σE. Models A and C perform acceptably, with R^2 well above zero, against the first dataset, but worse against the second, so they are not general enough. R^2 scores for Models B, D and E are very very close to 1 for both datasets.

Models A through E where created by adding improvements incrementally to the approximations of rate and number of actives, imposing restrictions on the available information the

978-1-5386-9156-4/18/$31.00 ©2018 IEEE
DOI 10.1109/eScience.2018.00081

TABLE I
MODEL PERFORMANCE (1ST DATASET)

Model	MAE	RMSE	R^2	μE	σE
A	0.79	4.25	0.84	0.52	4.22
B	0.26	1.46	0.98	0.19	1.45
C	1.91	6.89	0.59	-1.79	6.65
D	0.95	3.99	0.86	0.92	3.88
E	0.42	2.26	0.95	0.39	2.23

model had at a given moment and understanding the sources of inaccuracies and addressing these issues.

Model A

As was determined by in-depth analysis, the bandwidth available for transfers is not constant. But by assuming that it is constant during some time period however, it is possible to estimate the transfer rate. Model A uses the size of the transfers and the $started\_at$ and $transferred\_at$ timestamps to predetermine the rate based on active islands. The total volume of bytes transferred during this time is then divided by the number of active transfers.

An empirical search was conducted to find the scaling factor in the previous equation, and the best performance of the models over the first dataset was obtained with $f = 0.66$. We couldn't find a consistent explanation for this value in the observed data. This factor seems to be not constant as Model A behaves very well only in selected time spans. The reason why f is not 1, and not constant is unclear, and will be kept as a bias to be determined from historical data.

Model B

Doing post-mortem analysis it is possible to calculate the average rate of each transfer instead of the average of a group of transfers. The modifications to Model A involve adding a new predetermined rate to the simulated transfer structure. This rate is used at simulation time to make the transfer finish successfully.

This is the model that achieves the best results, but also the one that has the most information about the transfers available. Most of the data used to feed this model is not available at submission time though. Therefore, this model is not suitable to implement in online systems to make predictions about the transfers in real time, but only allows to validate the underlying model and is useful to make predictions about changes in the configuration of the underlying systems.

Model C

Model C replaces the rate calculation of Model B with a function, as described in Equation (1), which allows to predict the rate as a function of the size of the transfer. As this value is known at submission time, models that use this approach are suitable to be used in real time predictions of the number of queued transfers.

Model C however doesn't behave well when the bandwidth is underestimated. As some transfers take more time to finish in the simulation than in the observed data, the maximum number of actives used is the observed one, some new transfers could remain in the queue, where the maximum actives are zero, and these transfers will not exit until the next set of transfers trigger the max actives above zero. This problem had been seen in Model A and in with some insignificant effects, also seen in Model B.

Model D

To avoid queue starvation due to lack of actives the simplest solution is to limit the minimum max active transfers to 1, meaning if there are transfers in the queue at any given moment the simulator can take a transfer off the queue, and progress it. Model D uses this approach to approximate the maximum number of concurrent active transfers. This approach yields good results, comparable with Model B.

Model E

Model E uses a more sophisticated method to avoid starvation due to lack of active transfers. This method involves smoothing and shifting the observed number of concurrent actives. In some cases, the method outperforms the results obtained to Models B and D. Yet, overall, Models B and D are more general, simpler, and with comparable results, thus preferred over Model E.

Model B outperforms all the other models, with a R^2 score well above 0.95 and a RMSE which the lowest in both datasets.

The agreement between observations and predictions from Models D and E is also good according to R^2. These models only need the size of the transfers to approximate the rate. This study uses transfers exclusively from Chicago (MWT2) to Michigan (AGLT2). Accuracy of the models using other links needs to be studied. Inclusion of other observables, like number of transfers exiting the source or arriving the destination and the impact in the predictions, needs to be studied as well.

The work also present preliminary analysis of the power of the models to predict the time to complete of individual transfers. In these results, network time is predicted more accurately than queue time. More in-depth analysis is needed, but the preliminary results look promising.

REFERENCES

[1] Barisits, Beermann, Garonne, Javurek, Lassnig, Serfon, The ATLAS Data Management System Rucio: Supporting LHC Run-2 and beyond, ACAT, Seattle, 2017

[2] http://fts3-docs.web.cern.ch/fts3-docs/

[3] http://fts3-docs.web.cern.ch/fts3-docs/docs/optimizer/optimizer.html

[4] K. Jarrod Millman and Michael Aivazis. Python for Scientists and Engineers, Computing in Science & Engineering, 13, 9-12 (2011), DOI:10.1109/MCSE.2011.36

[5] Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

[6] John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95 (2007), DOI:10.1109/MCSE.2007.55