Operational Analytics Studies for ATLAS Distributed Computing: Data Popularity Forecast and Utilization of the WLCG Centers

Maria Grigoryeva[1,*], *Alexei* Klimentov[2,**], *Nina* Popova[1], *Mikhail* Shubin[1], and *Markus* Schulz[3], on behalf of the ATLAS Computing Activity[***]

[1]Lomonosov Moscow State University
[2]Brookhaven National Laboratory
[3]CERN

Abstract. Operational analytics is the direction of research related to the analysis of the current state of computing processes and the prediction of future states in order to anticipate imbalances and take timely measures to stabilize a complex system. There are two relevant areas in ATLAS Distributed Computing that are currently the focus of studies: user physics analysis including the forecast of popularity of data samples among users, and evaluating WLCG centers for their readiness to process user analysis payloads. Studying these areas is challenging due to the complexity involved, as it requires a comprehensive understanding of numerous boundary conditions typically found in large-scale distributed computing infrastructures. Forecasts of data popularity are problematic without the categorization of user tasks by their types (data transformation or physics analysis), which do not always appear on the surface but may induce noise, which introduces significant distortions for predictive analysis. Evaluating the WLCG resources by their analysis workloads is also a challenging task as it is necessary to find a balance between the workload of the resource, its performance, the waiting time for jobs on it, as well as the volume of jobs that it processes. This is especially difficult in a heterogeneous computing environment, where legacy resources are used along with modern high-performance machines. We will look at these areas of research in detail and discuss what tools and methods are used in our work, demonstrating results already obtained.

1 Introduction

Operational analytics, commonly referred to as operational intelligence, is the practice of utilizing data in realtime to make instant decisions in business operations. This refers to data that is collected and aggregated from existing business operations and then analyzed and fed back into operations instantly to make intelligent decisions on the spot. This study is devoted to the particular and highly topical subject of the operational analysis of the distributed computing system of the ATLAS Experiment at the LHC[1] – one of the largest experiments in

*e-mail: maria.grigorieva@cern.ch
**e-mail: alexei.klimentov@cern.ch

high-energy physics (HEP). Thus, operational analytics in this research is related to the analysis and forecasting of computing processes on the ATLAS distributed infrastructure, which uses the Worldwide LHC Computing Grid (WLCG)[2], in order to anticipate imbalances and take timely measures to stabilize a complex system.

There are two relevant areas in ATLAS Distributed Computing that are currently the focus of studies: user physics analysis including the forecasting of data popularity, and evaluation of WLCG centers to estimate their ability to process specific types of user analysis jobs. Both these topics are crucial for the efficient, stable and balanced utilization of the WLCG resources.

The first topic, the forecasting of data popularity[3], might lead to more optimal data management decisions. For example, anticipating the increase in the popularity of some scientific data in the next month might indicate the necessity of creating additional copies of this data at various computing centers to ensure better availability and faster processing. On the other hand, if some data is not expected to be accessed at all in the next few weeks, there is no need to add new copies, and the number of existing copies of this data might even be decreased. All these decisions and corresponding actions help to balance data volume and processing at the distributed resources.

The second topic, the assessment of the WLCG centers (sites), will become crucial for handling the expected user analysis payloads during the high luminosity (HL-LHC) era[4], when the LHC luminosity is increased by a factor of 5 to 7.5. Data volume and number of payloads will increase by 10x and this will lead to challenges for workload and data management systems. Ongoing studies describe the development of specific methods for the evaluation of the WLCG centers taking into account the diversity of resources and the chaotic nature of user analysis jobs.

In this research we demonstrate the current approaches to both these areas of the operational analytics of the distributed infrastructure of ATLAS experiment.

2 ATLAS Data Popularity Forecasting Studies

Data in the WLCG are organised in the form of datasets, which consist of files. A dataset has a specific format and belongs to a certain project. The popularity of a dataset is defined as a number of accesses to the dataset within a certain time period (day, week, month). In operational analytics, it is important to take into account not only individual datasets but also groups of datasets that share a common format and project. The popularity of a group of datasets is determined by the total number of accesses to the datasets in that group. Thus, the popularity of a group of datasets represents a time series with weekly calculated number of accesses to the group of datasets. For this research a dataset group of Derived Analysis Object Data (DAOD)[1], belonging to the mc16_13TeV project[2] and PHYS format description[3] was chosen. The total length of time interval under investigation was about 2.5 years. The number of datasets included in the study group was 30 thousands.

A large number of methods have been developed for time series analysis and forecasting, including statistical, machine learning (ML) and hybrid algorithms. With respect to the problem to be solved, forecasting can be considered as a regression task. For prediction, both basic classical ML methods such as Logistic Regression [5], Random Forest [6], etc., and

[1]DAOD is the format optimized for analysis and contains objects and information relevant to specific analyses

[2]mc16_13TeV project contains Monte-Carlo simulated events corresponding to the run 2 data-taking campaign for pp collisions with an energy of 13 TeV

[3]PHYS is the common derivation format intended for 80% of all physics analyses, not including non-standard analyses, like long-lived particles or signatures, custom jet collections

more complex models such as recurrent neural networks based on LSTM (Long Short-Term Memory) [7, 8] or GRU (Gated Recurrent Unit) [9] may be used.

This work considers several machine learning methods as examples for the time series forecasting: one-layer LSTM, stacked LSTM, Random Forest.

2.1 Time Series Preprocessing

Before applying each of the proposed models, the time series is smoothed by moving average with window size of one week. Then a logarithmic transformation: Log(x + 1) is applied to the smoothed time series.

Such a filtering is applied due to the presence of a large number of local minima and maxima in the original time series, which are difficult for models to handle.

To make the time series stationary an extra preprocessing stage is needed. For One-Layer LSTM and Random Forest we used differencing: the difference between the current value and the previous value for every time moment. For stacked LSTM min-max scaling to $[-1, 1]$ segment was used.

2.2 One-layer LSTM Forecast

LSTM[7, 8] is a recurrent neural network, that takes a sequence of (possibly vector) values and returns one (possibly vector) value or a sequence of the same length. It has a hidden state cell and four layers of weights with sigmoid and tanh activation.

The first considered model consists of one LSTM layer and one Dense layer with 1D max-pooling layer between them. The input to the LSTM consists of n values of the sequence preceding the ones that need to be predicted (n_lags). The dimension of the LSTM hidden state (n_units) was chosen to be 24 as a trade-off between model complexity and prediction time. LSTM outputs the whole sequence, that is then passed to 1D max-pooling along the time axis. The result is then fed into a fully connected layer of size n_units × n_forecasts, where n_forecasts is the number of predicted values in the future. Thus the neural network gives several future values simultaneously as a result (so-called *multi-output prediction*).

2.3 Extended Neural Network: Stacked LSTM

A more complex model is a neural network consisting of several LSTM layers. In such a network, the first LSTM layer takes the input time series (a sequence of scalar values), it has shape $(l, 1)$, where l is sequence length, "1" stands for one-dimension data. The layer then outputs a sequence of vector values of some dimension d_1, which equal to LSTM's hidden state dimension, so the sequence has shape (l, d_1). This sequence is passed to the next LSTM layer as input and so on. This architecture is called *stacked LSTM*.

Stacking multiple LSTM layers makes the model deeper and more complex, so that it is able to create new representations at high levels of abstraction. Such models have been applied to speech recognition tasks [10].

In our case it is useful to have several hidden LSTM layers with dimensions bigger than one, but the output layer should be of dimension one, as the model should predict scalar values.

In this work the architecture of the extended model consists of five LSTM layers with dimensions of 64, 128, 128, 64, and 1. Model training requires passing the whole time series through the network (as one training object) multiple times. The correct answer is the same time series shifted by the number of predicted steps.

2.4 Random Forest Forecast

Random Forest is a classical machine learning algorithm for solving classification and regression problems. The method consists of constructing an ensemble of decision trees, each considering a different subset of features.

In this work, the Random Forest is used in the form of autoregressive forecasting [11]. The training objects are formed from the time series values, suche that for each value, n_lags previous values of the series become the features. To predict several values ahead a *multistep mode* is used, where every next predicted value is added as a feature to a new object. The hyperparameters of the method are the maximum tree depth and the number of trees in the forest; in this paper they were chosen to be 3 and 1000 respectively.

2.5 Cross Validation of Proposed Methods and Naive Forecast

To evaluate the proposed methods, we carried out a cross validation and investigated the relationship between learning quality and the length of the training time series. A *Naive Forecasting* method was added to the observed methods. The Naive method outputs shifted values as predictions. Cross validation was performed in a block mode: fixed-length blocks were selected from the time series, each block consisting of a training part and a test part for prediction. The block beginnings were selected with a fixed step, and the blocks could overlap. The models were evaluated on the transformed data (with *moving average smoothing* and logarithm). On the original data the prediction accuracy is significantly lower. As mentioned above, the models cannot cope with rapid oscillations and peaks. After the *moving average smoothing* the time series continues to have a big variance, so the additional logarithmic transformation is used.

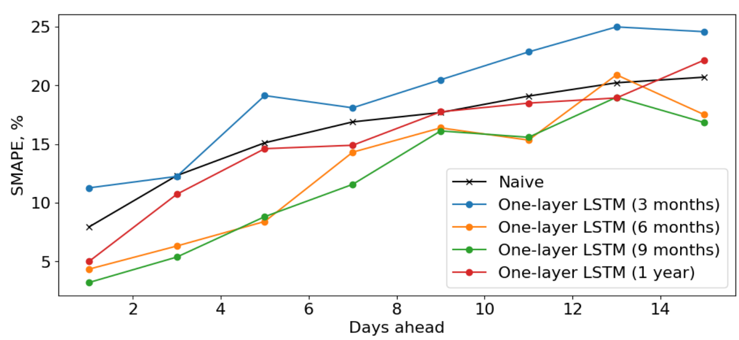

Figure 1. Cross validation of One-layer LSTM compared to Naive forecast for different training time series length on PHYS datasets group. The results show SMAPE metric depending on prediction horizon

Figures 1, 2, 3 present the cross validation results for One-layer LSTM, Stacked LSTM and Random Forest models. They show the Symmetric Mean Absolute Percentage Error (SMAPE)[4] metric depending on the prediction horizon (days ahead) for different lengths of the training time series. The prediction horizon varies from 1 to 15 days, while training time series length changes from 3 months to 1 year. Each of these three plots includes the same Naive Forecasting result.

[4]SMAPE quantifies the percentage difference between the observed and predicted values

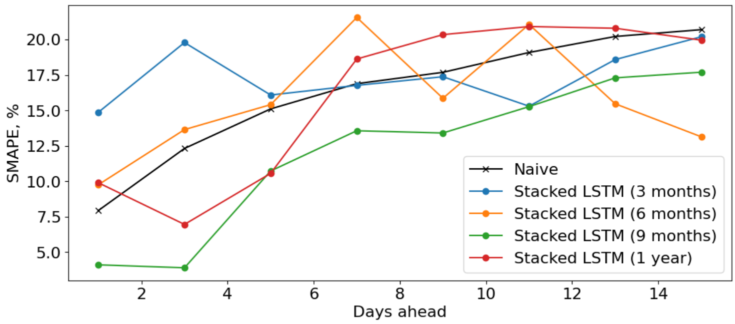

Figure 2. Cross validation of Stacked LSTM compared to Naive forecast for different training time series length on PHYS datasets group

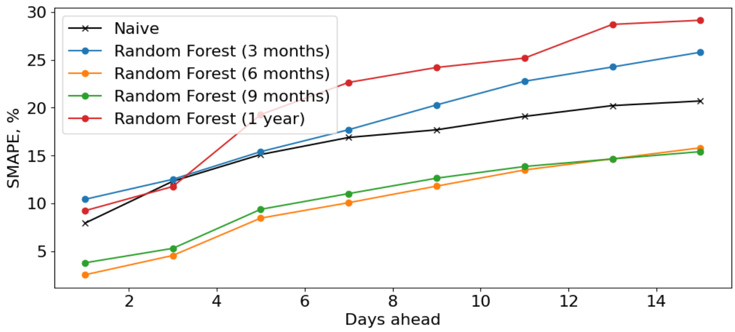

Figure 3. Cross validation of Random Forest compared to Naive forecast for different training time series length on PHYS datasets group

It can be seen that some configurations of methods provide consistently better prediction than the Naive Forecast: One-layer and Stacked LSTMs trained on 9 months of data, Random Forest trained on 6 months and on 9 months. The increase of prediction horizon leads to an increased error in almost all cases.

Stacked LSTM has the most unstable results, where only the 9 months trained model surpasses Naive Forecast. Random Forest versions trained on 6 and 9 months showed better predictions than LSTM models.

While the developed models are effective in forecasting logarithmic smoothed data, they are not precise in predicting the exact number of dataset accesses on particular days. However, they can still be utilized to forecast overall trends at a coarse-grained scale of a week.

As future work directions, it is proposed to consider analyzing the popularity of individual datasets taking into account additional features besides the number of tasks accessing the dataset.

3 WLCG Centers Analysis Workload Evaluation

The distributed, heterogeneous infrastructure used by the ATLAS experiment is the largest computing grid in the World. Over 176 computing centres, distributed all over the world, contribute resources. Up to 5 000 users utilize these resources for physics analysis, and the data volume has recently exceeded 1 exabyte. Thus, we deal with very large multi user infrastructure where millions of payloads run daily. Evaluation of user analysis payloads of a WLCG resource requires a relative metric showing the ability of a resource to process user analysis jobs in comparison to all other resources. The complexity of the assessment is in the specificity of the WLCG and its distributed nature.

In this research we are focused on the assessment of the resources based on their real historical user analysis payloads in order to exploit this information to anticipate how these payloads will be processed in future. We propose the following metric of WLCG resources: by the time that jobs spend in waiting, running and finalizing states.

Jobs statuses were categorized into three types: waiting (assigned, activated, defined, pending, and other waiting statuses in a queue, including time to download input files to a worker node), running, and finalizing (statuses related to finishing jobs, such as merging or transferring).

We calculated the average proportion of waiting, running, and finalizing time jobs spent within an hour at efficient and stable ATLAS-CERN and ATLAS-US computing centers over a one-month period. Approximately half the time jobs spent running, while around 20-25% of an hour they spent in waiting and finalized states. ATLAS-CERN and ATLAS-US represent regional groups of resources, each consisting of multiple sites. The proportions can be expressed using the following formula (Equation 1), where U_t represents workload and $\overline{T_{jw}}$, $\overline{T_{jr}}$, and $\overline{T_{jf}}$ denote average waiting, running, and finalizing times respectively:

$$U_t = \frac{\overline{T_{jw}} + \overline{T_{jf}}}{\overline{T_{jr}}} \tag{1}$$

To avoid the contribution of failed payloads, in this research only successfully finished jobs were taken into account. It should be noted that all evaluations were conducted on distributed queues associated with ATLAS sites. Each site comprises multiple queues, each serving a specific purpose. Figure 4 demonstrates the evaluated analysis workload of three ATLAS-CERN sites over one randomly chosen week. There is obviously one resource that was running not many analysis jobs — ANALY_CERN_T0_ART, the workload at CERN resource is normal, and at CERN-T0 it is normal as well, with short periods when this resource seems to be overloaded with analysis jobs.

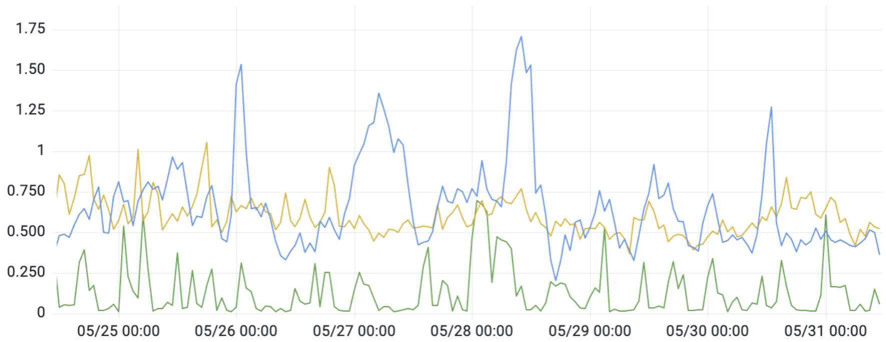

Figure 4. User analysis workload at three ATLAS-CERN grid sites over a period of one week

3.1 User Analysis Workload Panel of the Resources Dashboard

Based on the metric developed, a special dashboard was implemented for the ATLAS MONIT infrastructure[13]. This dashboard has multiple panels for the monitoring of the analysis workload of the WLCG resources by various parameters, but the most representative ones are the current (aggregated for the last 6 hours) workloads of the resources by groups. See Figures 5, 6), where waiting_time corresponds to $\overline{T_{jw}}$, running_time — $\overline{T_{jr}}$ and finalizing_time — $\overline{T_{jf}}$. The analysis workload of all WLCG clouds is shown in Figure 5, and sorted in an

cloud	waiting_time	running_time	finalizing_time	analysis_workload	n_jobs
CERN	9.13	33.3	12.3	0.270	13.5 K
FR	9.11	30.6	12.0	0.300	17.7 K
RU	9.71	31.3	13.4	0.310	2.22 K
US	11.2	32.9	12.9	0.340	44.6 K
ES	13.0	29.4	11.2	0.440	4.32 K
IT	13.1	27.6	11.5	0.470	6.64 K
DE	14.3	28.4	10.8	0.500	21.7 K
CA	15.0	28.7	11.8	0.520	11.1 K
UK	14.3	27.2	9.89	0.520	16.0 K
TW	12.8	23.1	7.44	0.740	1.16 K
NL	19.4	25.8	11.0	0.750	9.56 K
ND	20.8	23.1	11.7	0.890	16.7 K

Figure 5. Current user analysis workload of ATLAS clouds [time is in minutes per job].

cloud	waiting_time	running_time	finalizing_time	analysis_workload	n_jobs
RAL-LCG2_MCORE	60	0	0		1
UKI-LT2-RHUL_VP	2.92	30.6	6.81	0.100	205
UKI-NORTHGRID-MAN-HEP	8.05	34.5	11.5	0.230	2.44 K
RAL	10.7	33.3	12.5	0.320	7.53 K
UKI-LT2-RHUL	8.67	21.2	3.78	0.400	271
UKI-SOUTHGRID-RALPP	9.42	23.0	6.72	0.410	167
UKI-NORTHGRID-LANCS-HEP-CEPH	9.05	21.7	8.33	0.420	900
UKI-LT2-QMUL	14.8	29.7	12.1	0.500	2.15 K
UKI-SCOTGRID-GLASGOW_CEPH	10.6	19.1	4.94	0.510	136
ANALY_UKI-SOUTHGRID-OX-HEP_VP	8.84	13.6	0.0667	0.520	182
UKI-NORTHGRID-LIV-HEP	28.0	19.1	5.47	0.920	114
UKI-SCOTGRID-ECDF	19.2	17.5	5.74	1.13	216
UKI-SOUTHGRID-BHAM-HEP_VP	21.8	17.2	4.81	1.21	277
UKI-NORTHGRID-SHEF-HEP	6.99	2.24	0.0667	2.68	33

Figure 6. Current user analysis workload of queues in the ATLAS-UK cloud [time is in minutes per job].

ascending order from resource least loaded with analysis jobs, which is ATLAS-CERN, to normally and overloaded clouds. The average number of jobs executed per 6 hours (n_jobs), calculated as the total number of queued, running and finalizing jobs, is also shown as it ensures the measurement of the average capacity of all clouds. When the brokerage distributes jobs by resources, besides the mapping of these jobs to the computing nodes or queues, the proportions of jobs being sent to each resource must also be calculated. In this case the capacity of the resources might be taken into account. The greater the average number of jobs a resource handles per hour, the more new jobs can be sent to it.

From this table, we can conclude that job waiting time contributes a lot to the analysis workload assessment. But if we look more closely at the metric of more specific entities, like queues, it is obvious how all three time metrics contribute to the final value. Figure 6 demonstrates the analysis workload of ATLAS-UK sites, and the most overloaded one has relatively short waiting time, but the duration of running and finalizing jobs is even smaller, which indicates that there are more waiting processes than executing at this resource.

4 Conclusion

The paper presents two research directions in operational analysis of ATLAS Computing Activities.

In terms of data popularity forecasting different methods, such as LSTM, Stacked LSTM, and Autoregressive Random Forest, were tested for assessing dataset popularity. Moving average and log-transformation helped generate accurate forecasts up to one week in advance. Our models outperformed Naive Forecast when trained on 6 and 9 months of data. The Random Forest model proved to be the most stable. Our conclusions are supported by the results of the conducted tests, as shown in Figures 1 and 3. The green and orange lines corresponding to the forecasts based on 6 and 9 months are below the black line corresponding to the Naive prediction, indicating that we are achieving a lower error. In the case of Stacked LSTM (Figure 2), we were only able to outperform the Naive prediction at 9 months. The models trained on 3, 6 months, and a year turned out to be comparable or even slightly worse than the Naive Forecast. A target accuracy of 5% SMAPE can be considered for the predictions. Therefore, our current models can forecast data popularity trends for a horizon of 1-3 days.

An advanced job calculation method was proposed to estimate the workload analysis of WLCG resources. This method enables the estimation of jobs at PanDA queues within 1-hour intervals. An analysis workload dashboard was implemented in the ATLAS experiment monitoring infrastructure, displaying resource analysis workload for the last 6 hours in the WLCG. Validation of the metric requires further research comparing existing workload management decisions with those suggested by resource ranking algorithms.

References

[1] ATLAS Collaboration, “The ATLAS Experiment at the CERN Large Hadron Collider,” JINST **3**, S08003 (2008)

[2] I. Bird, K. Bos, N. Brook, D. Duellmann, C. Eck, I. Fisk, D. Foster, B. Gibbard, C. Grandi and F. Grey, *et al.* “LHC computing Grid. Technical design report,” CERN-LHCC-2005-024.

[3] T. Beermann *et al.*, "Methods of Data Popularity Evaluation in the ATLAS Experiment at the LHC," EPJ Web Conf. **251**, 02013 (2021)

[4] J. Bendavid, “High Performance Analysis, Today and Tomorrow,” J. Phys. Conf. Ser. **2438**, no.1, 012002 (2023)

[5] D. Kleinbaum et al., *"Logistic regression"* (Springer-Verlag, New York, 2002)

[6] L. Breiman, "Random forests", Machine learning **45**, 5–32 (2001)

[7] F. Karim et al., "Multivariate LSTM-FCNs for Time Series Classification", arXiv:1801.04503 (2018)

[8] F. Karim et al., "Insights Into LSTM Fully Convolutional Networks for Time Series Classification", IEEE Access **7**, 67718–67725 (2019)

[9] M. Rahman et al., "Predicting Prices of Stock Market using Gated Recurrent Units (GRUs) Neural Networks", International J. of Computer Science and Network Security **19** (1), 213–222 (2019)

[10] A. Graves et al., "Speech Recognition with Deep Recurrent Neural Networks", IEEE International Conference on Acoustics, Speech and Signal Processing **38**, 6645–6649 (2013)
[11] S. Seitz, "Forecasting with Decision Trees and Random Forests", URL: https://www.sarem-seitz.com/forecasting-with-decision-trees-and-random-forests/. Accessed 2023
[12] ATLAS Collaboration, “Evolution of the ATLAS PanDA workload management system for exascale computational science,” J. Phys. Conf. Ser. **513**, 032062 (2014)
[13] A. Aimar, A. Aguado Corman, P. Andrade, J. Delgado Fernandez, B. Garrido Bear, E. Karavakis, D. M. Kulikowski and L. Magnoni, “MONIT: Monitoring the CERN Data Centres and the WLCG Infrastructure,” EPJ Web Conf. **214**, 08031 (2019)