# Predictive analytics tools to adjust and monitor performance metrics for the ATLAS Production System

F.H. Barreiro Megino[1], M. Borodin[2,4], D. Golubkov[3,4], M. Grigorieva[4,5], M. Gubin[5], A. Klimentov[4,6], T. Korchuganova[5], T. Maeno[6], S. Padolski[6], M. Titov[4]
on behalf of the ATLAS Collaboration

[1] University of Texas at Arlington (US), [2] University of Iowa (US), [3] Institute for High Energy Physics (RU), [4] National Research Centre «Kurchatov Institute» (RU), [5] National Research Tomsk Polytechnic University (RU), [6] Brookhaven National Laboratory (US)

ACAT 2017
21-25 August 2017
University of Washington, Seattle

## Abstract

Having information such as an estimation of the processing time or possibility of system outage (abnormal behaviour) helps to monitor system performance and to predict its next state.

The current cyber-infrastructure presents computing conditions in which contention for resources among high-priority data analysis happens routinely, that might lead to significant workload and data handling interruptions. The lack of the possibility to monitor and to predict the behaviour of the analysis process (its duration) and system's state itself caused to focus on design of the built-in situational awareness analytic tools.

## ATLAS Production System

The ATLAS **Prod**uction **Sys**tem (ProdSys) [1] is an automated scheduling system that is responsible for central production of Monte-Carlo data, highly specialized production for physics groups, as well as data pre-processing and analysis using such facilities as grid infrastructures, clouds and supercomputers. Its current processing rate is around 2M tasks per year which is more than 365M jobs per year.

ProdSys evolves to accommodate a growing number of users and new requirements from the ATLAS Collaboration, physics groups and individual users. ATLAS Distributed Computing in its current state is the aggregation of large and heterogeneous facilities, running on the WLCG, academic and commercial clouds, and supercomputers.

**Core components** [2]

- Database Engine for Tasks (DEfT)
  formulate the tasks, chains of tasks and task groups (i.e., production request), complete with all necessary parameters
- Job Execution and Definition Interface (JEDI)
  task-level workload management (i.e., brokerage and execution), dynamic job definition and execution (for resources usage optimization)


**Figure 1.** ProdSys workflow.

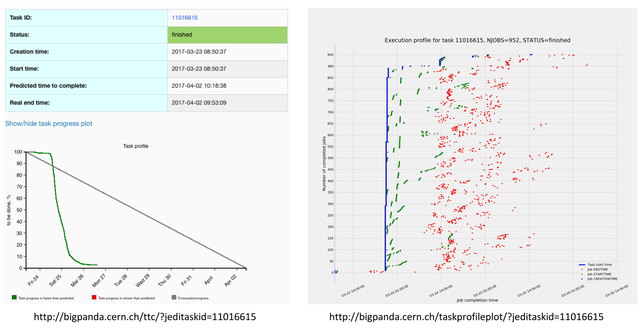


**Figure 2.** Task profile @BigPanDA Monitor (taskID=11016615).

## Overview

The proposed suite of tools aims to estimate completion time (so called "Time To Complete", TTC) for every (production) task, completion time for a chain of tasks, and to predict the failure state of the system based on "abnormal" task processing times. Its implementation is based on Machine Learning methods and techniques, and besides the historical information about finished tasks it uses ProdSys job execution information and resources usage state (real-time parameters and metrics).

The WLCG ML R&D project started in 2016. Within the project the first implementation of the TTC Estimator (for production tasks) was developed, and its visualization was integrated into the ProdSys Monitor.

## Methods and Techniques

Apache Spark [3] data processing framework and Spark.Mllib [4] machine learning library are used as a foundation for the prediction system.

**Ensemble methods** *(ensemble of decision trees)*

- Gradient-Boosted Trees regression method
- Random Forests regression method

**Prediction models**

- Cold – based on task definition parameters that categorize the average execution process for the defined task type (with particular conditions). Gives task duration estimation during its definition.
- Warm – based on description and state of scout jobs (i.e., scouts) that are used to check the processing environment. Gives prediction for the task duration immediately after task launch.
- Hot – based on the current state of task processing (states of environment and corresponding jobs). Gives prediction adjustment during the task execution.

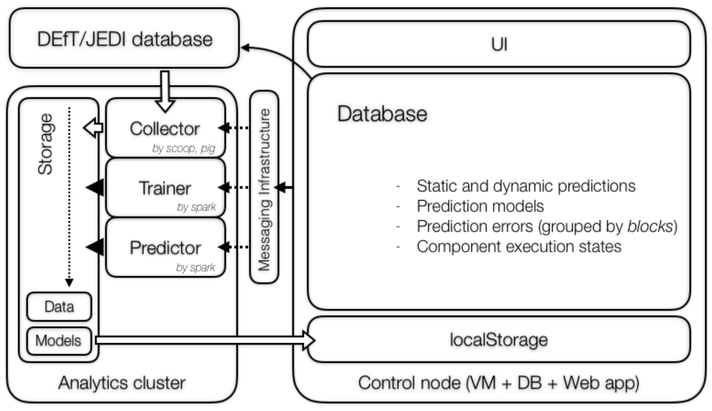


**Figure 3.** ProdSys Predictive Analytics workflow.

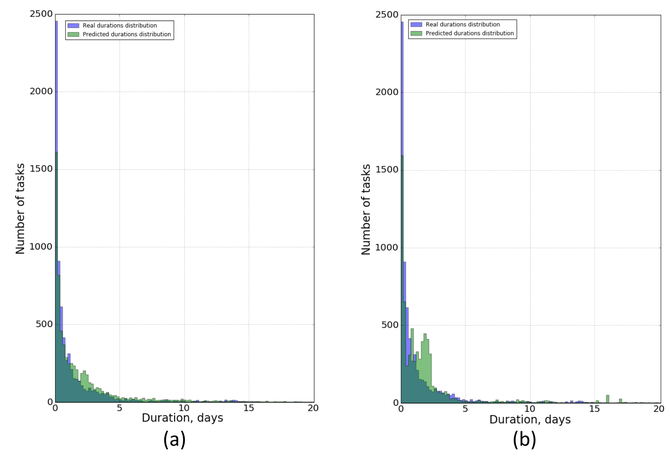


**Chart 3.** GBT (a) and RF (b), task duration distribution (real vs predicted).

## Gradient-Boosted Trees

- Advanced model tuning: number of trees, depth (or number of leaves), and the shrinkage (or learning rate) - affects the model behavior;
- High performance while dimensionality is low;
- Robust when the training set contains outliers;
- More prone to overfitting than RF;
- Outperform random forests when properly tuned, give worse results otherwise.

Example (charts 1,3): training data - 3 months, test data - 1 month
absolute error (days): mean = 1.05, std = 4.59, 3 sigma = 13.79
RMSE = 4.72 days

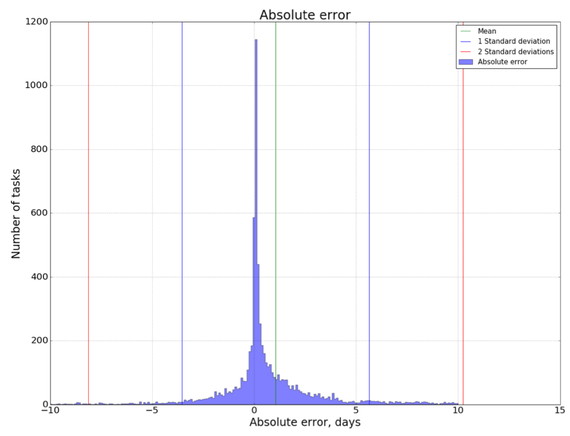


**Chart 1.** GBT, absolute error.

## Random Forests

- Simple model tuning: only tunable parameter is the number of features to randomly select at each node;
- Less prone to overfitting than GBT;
- Marginally better performance of parallel implementations;
- Robust when the training set contains outliers.

Example (charts 2,3): training data - 3 months, test data - 1 month
absolute error (days): mean = 0.63, std = 2.57, 3 sigma = 7.72
RMSE = 2.65 days

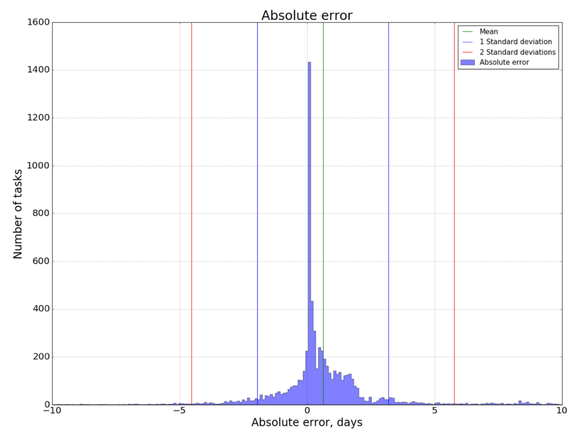


**Chart 2.** RF, absolute error.

## Conclusions

Adding task duration estimation to the features of the ProdSys monitoring will allow detection of task execution delays as they happen, and take measures to resolve them. Combined with the anomaly detection framework, this will further improve efficiency of ProdSys data processing and reduce the data handling interruptions.

## References

1. M. Borodin et al., "Scaling up ATLAS production system for the LHC Run 2 and beyond: project ProdSys2," J. Phys. Conf. Ser., vol. 664, no. 6, 2015.
2. K. De, D. Golubkov, A. Klimentov, M. Potekhin and A. Vaniachine, "Task Management in the New ATLAS Production System," J. Phys. Conf. Ser., vol. 513, no. 3, 2014.
3. https://spark.apache.org
4. X. Meng et al., "MLlib: machine learning in apache spark," The Journal of Machine Learning Research, vol.17 no.1, pp.1235-1241, 2016.