Journal of Physics: Conference Series **664** (2015) 032003 doi:10.1088/1742-6596/664/3/032003

Exploiting CMS data popularity to model the evolution of data management for Run-2 and beyond

D Bonacorsi[1], T Boccali[2], D Giordano[3], M Girone[3], M Neri[1], N Magini[3], V Kuznetsov[5], T Wildish[6]

[1] University of Bologna
[2] INFN Pisa
[3] CERN
[4] Fermilab
[5] Cornell University
[6] Princeton University

E-mail: Daniele.Bonacorsi@bo.infn.it

Abstract.

During the LHC Run-1 data taking, all experiments collected large data volumes from proton-proton and heavy-ion collisions. The collisions data, together with massive volumes of simulated data, were replicated in multiple copies, transferred among various Tier levels, transformed/slimmed in format/content. These data were then accessed (both locally and remotely) by large groups of distributed analysis communities exploiting the WorldWide LHC Computing Grid infrastructure and services. While efficient data placement strategies - together with optimal data redistribution and deletions on demand - have become the core of static versus dynamic data management projects, little effort has so far been invested in understanding the detailed data-access patterns which surfaced in Run-1. These patterns, if understood, can be used as input to simulation of computing models at the LHC, to optimise existing systems by tuning their behaviour, and to explore next-generation CPU/storage/network co-scheduling solutions. This is of great importance, given that the scale of the computing problem will increase far faster than the resources available to the experiments, for Run-2 and beyond. Studying data-access patterns involves the validation of the quality of the monitoring data collected on the "popularity of each dataset, the analysis of the frequency and pattern of accesses to different datasets by analysis end-users, the exploration of different views of the popularity data (by physics activity, by region, by data type), the study of the evolution of Run-1 data exploitation over time, the evaluation of the impact of different data placement and distribution choices on the available network and storage resources and their impact on the computing operations. This work presents some insights from studies on the popularity data from the CMS experiment. We present the properties of a range of physics analysis activities as seen by the data popularity, and make recommendations for how to tune the initial distribution of data in anticipation of how it will be used in Run-2 and beyond.

1. Introduction

The CMS experiment [1] at the LHC accelerator is concluding the first Long Shutdown (LS1) after a successful first run of data taking (Run-1), and Run-2 is currently expected to start in Summer 2015. Plenty of data have distributed world-wide using a CMS-specific data management system that runs on top of Grid transfer tools, called PhEDEx [2, 3] (more than

150 PB moved during Run-1, currently moving about 2.5 PB per week among about 60 Grid sites). The distributed community of physicists is accessing Run-1 data and simulated data using the CRAB system [4, 5], currently at version 3. Since few years, CMS is collecting data on the most frequently datasets being accessed at all Grid sites of the Worldwide LHC Computing Grid (WLCG) [6, 7], thus enabling CMS to study the so-called "popularity" of each sample (more details in [8]). This information is collected in quite some details: when one of the replicas of each CMS dataset is accessed via CRAB analysis jobs at a certain computing sites, the timestamp of the access, the site hosting the data, the number of existing replicas, the location of such replicas on the computing infrastructure, the amount of CPU time used to access such dataset, etc. are collected and saved in a central DB for further aggregation and consumption. One of the first impact of such effort on the CMS Computing operations is that the CMS data placement is evolving towards a less static model, i.e. replicas are added (removed) for datasets that appear to be most (least) popular. This activity is already in production for CMS and documented in this same conference [9]. In this paper we discuss a complementary, forward-looking approach. The basic idea is to profit from the experience that can be gained from studying the access pattern to the data in the present to perform prediction for the data accesses in the near future, namely predict which datasets will become popular once they will be available on the Grid for distributed analysis.

2. The CMS Analytics approach

The CMS experiment recently launched a project focussed on Data Analytics techniques. By "data" here we refer to any (meta-)data produced by any Computing Operations activity since the start of Run-1. The project has different goals depending on the timeline. As a long-term goal (2-3 years), the project aims to build an adaptive data-driven model of the CMS Data Management (DM) and Workload Management (WM) activities [10], with the target of being able to predict future behaviours of the CMS computing systems in operations from the measurements of their past performances. A medium-term goal (e.g. within or soon after Run-2), the project aims to use the information coming from analytics studies to improve the use of CMS computing resources (e.g. optimise only the disk occupancy, minimise the number of datasets' replicas, etc); this is a remarkable goal in itself, but smaller in scope with respect to the previous one. As a short-term goal (within Run-2) instead, the project aims to just provide a concrete support the CMS Computing Operations team through a deeper understanding and analysis of the data that describe the way CMS Computing operations were performed in Run-I and LS1, at least in some sectors, with priorities driven by Computing Operations needs. The rationale behind this modelling to be adaptive is that models that were elaborated in the past are not going to apply to the future for long, thus only really adaptive modelling would equip CMS with predictive power in the long term.

The CMS Analytics project is subdivided in a number of sub-projects. We proceed by adding all possible ideas in a pool of potentially interesting projects which are brainstormed and studied for feasibility and in terms of potential impact; all of them are thought to be self-contained and well-defined in scope and timeline. In order to protect the limited manpower we have, only once a sub-project is found to be promising and useful for CMS Computing Operations as a whole, the sub-project is actually started and pursued until completion. In the following, the data treatment is discussed in general, and first observations from the application of this approach on the particular case of the "data popularity" are presented.

3. Data treatment

The operations model built by CMS before Run-1 worked successfully and met all requirements. Despite it was a success and well served the CMS physics program according to all possible metrics, in preparation for Run-2 (and beyond) most systems underwent a review to assess

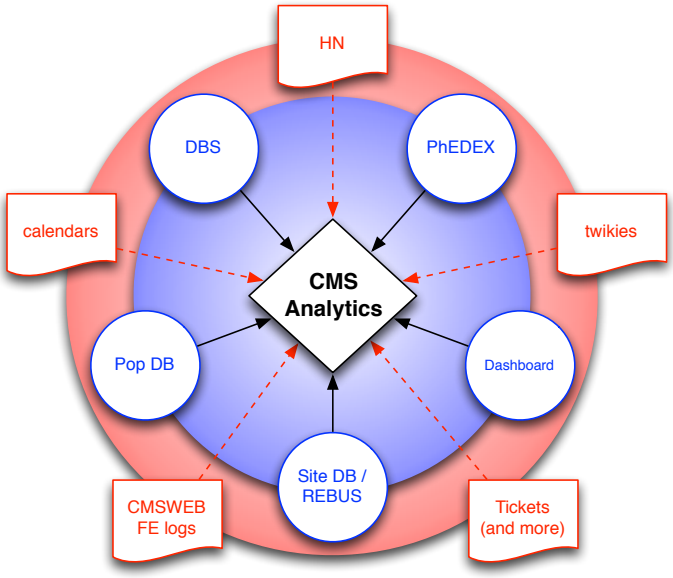

Figure 1. Different sources of structured and unstructured data to be fed to the CMS Analytics project.

their readiness for the next data taking round. In general, a lesson learned was that despite a given system worked, but it ended up in not being completely understood in the way it actually operated over several years. As an example, the data traffic with Tier-2 sites as destinations exceeded the expectations as from the MONARC model [11], and we found no way to reasonably shape this transfers. As another example, the disk storage at the Tier-2 level was filled with AOD/AODSIM data tiers, but plenty of it was not accessed by distributed analysis jobs over long periods of time, thus accounting for not efficient resource utilisation.

In the CMS Computing operations in Run-1 and LS1, all computing systems collected large amounts of data about operations themselves, namely monitoring data, accounting information, machine logs, etc. This data refers to transfers, job submissions, site efficiencies, release details, infrastructure performance, analysis throughput, and much more. All this data is archived, but never really accessed and analysed a-posteriori by anyone - as we mainly monitor our systems in near-time for debugging purposed only, rather than analyse a large bunch of such data to study in depth the behaviour of a complex system. Additionally, the data as such is neither complete nor validated, thus resulting in a data set that needs to go under a data preparation and validation step before being suitable for any further analysis. Once this cleaning and validation is performed, this data stands as a precious input for a Big Data analytics project. In terms of the four big V's that define Big Data, some apply to this context more than others. The Volume (scale of the data) is not negligible in itself, but definitely manageable with respect to the total volume of the LHC collisions data we deal with, or the large amounts of simulated data. The Velocity (analysis of streaming data) is partially relevant, in the sense that we may aim to a quick availability of analytics results, but having a real-time system is not actually a requirement. The Variety (different forms of data) is very relevant, as we deal with a very irregular data set, consisting of structured, semi-structure and unstructured data (as explained in the following). The Veracity (uncertainty of data) is also delicate, as the integrity of the data and the ability to trust the outcome of the data analysis to make informed decision is very critical.

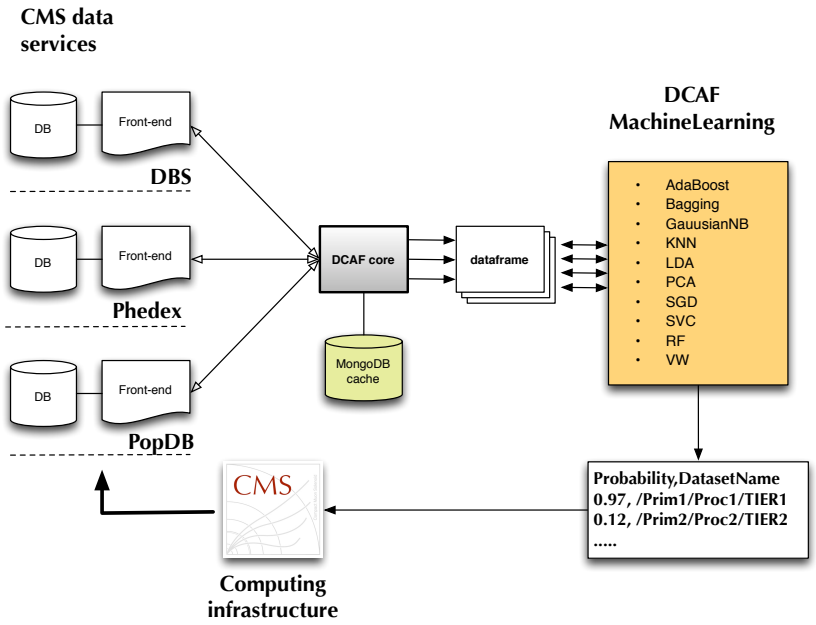

Figure 2. Description of the DCAFPilot, in terms of workflow and components.

3.1. Structured and unstructured data

Many CMS Computing activities in their execution produce data which are stored across multiple data services and are available via CMS data service APIs: all these information are "structured" in type (see Fig. 1, blue color). Some examples can be listed: the DBS system (the CMS source for physics meta-data); the PhEDEx transfer management database (central database for CMS data transfer tasks, with data replica catalog functionalities); the Popularity Database (it collects dataset user access information); SiteDB (it collects authoritative information about pledges at WLCG sites, deployed resources, as well as manpower onsite); the CERN Dashboard (a large repository of details e.g. on Grid jobs, etc).

In addition to structured data, much data in the CMS Computing ecosystem is partially or completely unstructured (see Fig. 1, red color). With respect to the former kind, the latter is much harder to collect and process, but potentially extremely rich in content. Some examples of unstructured data are listed in the following, and some more details about what they may offer can be found in [12]. The CMS HyperNews system hosts more than 400 different fora for CMS today, and it represent de-facto a reference on several years of user activities, covering e.g. announcements, hot topics, migration of interests in the user communities, etc. The ticketing systems (Savannah, GGUS, but also discussions on activity-based ELOGs or topical e-groups) give an overview of infrastructure issues and sites status over large windows of time. The CMS portion of the CERN-based twiki system contains plenty of information on various CMS software/computing topics, a sort of knowledge graph that could be mapped to user activities and physics interests and may be used to model their evolution over the years. The CMS calendar (including the list of major conferences and workshops that hosted CMS speakers) could be used to track hot periods in CMS physics analysis, and may be used to identify seasonal cycles in each physics community (also Vidyo logs could be helpful to pursue this goal). Even semi-structured data like the *cmsweb* cluster front-end logs could be valuable to extract information on user activities.

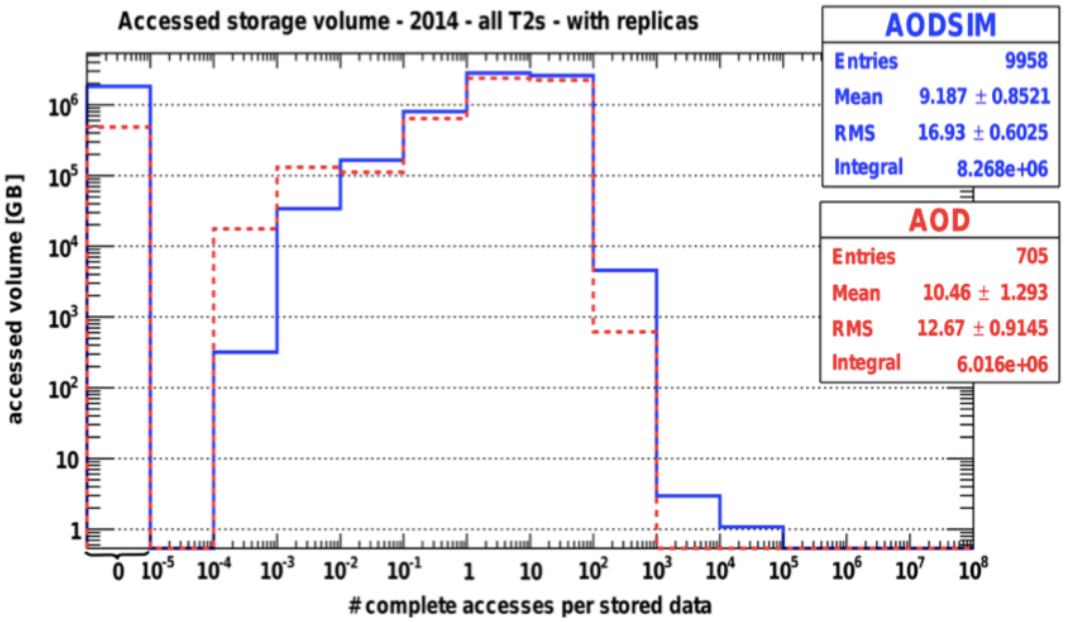

Figure 3. The global CMS Tier-2 storage volume hosting AOD/AODSIM accessed in 2014, as a function of the number of complete accesses per stored data (note the bi-logarithmic scale).

4. A use-case: CMS data popularity

The data described in the previous section, if properly handled and mined, has the potential to act as a data vault on which predictions for specific observables may be tried. In this section we describe how the "data popularity" can be of interest in this task, a the details of a pilot project to demonstrate it.

As stated in the introduction, the CMS Dynamic Data Placement team relies on historical information on the popularity of CMS datasets to add (remove) replicas of existing datasets that are most (least) desired by end-users. This approach, demonstrated to be useful in production, has one flaw though: it reacts to spikes in the dataset popularity only after the fact. It is desirable to predict which datasets *will* become popular even *before* such datasets are available on Grids.

The data popularity, despite relatively simple in definition, is a complex observable once it is deeply investigated in all its aspects and if it is studies in its evolution over time for a large set of data like all the data available on WLCG sites to CMS analysis users. In Figure 3, the accessed storage volume to AOD/AODSIM data format on all CMS Tier-2 centers in the entire 2014 is displayed as a function of the number of complete accesses per stored data. As the plot shows, the bulk of the accesses is not on a single peak, but actually quite dispersed. But one bin is the most striking, i.e. the first one 'artificially' added in correspondence to 0 accesses (it will be called "bin-0" in the following). This bin quantifies an important observation: in CMS Computing operations in 2014, 16% (29%) of the total storage at Tier-2 sites occupied by AOD (AODSIM) has been accessed not even once in the entire 2014. This information is completely lost in case a blind average is done to compute the mean number of complete accesses per stored byte. E.g. studying this in January 2015 and looking at different time windows in the past (1 month, 3 months, 6 months, etc), it could be easily extracted for example that in 2014 each bit written on Tier-2's was accessed on average about 10 times: a decent and reassuring figure, which on the other hand averages on accessed and not accessed data though, so it actually fails

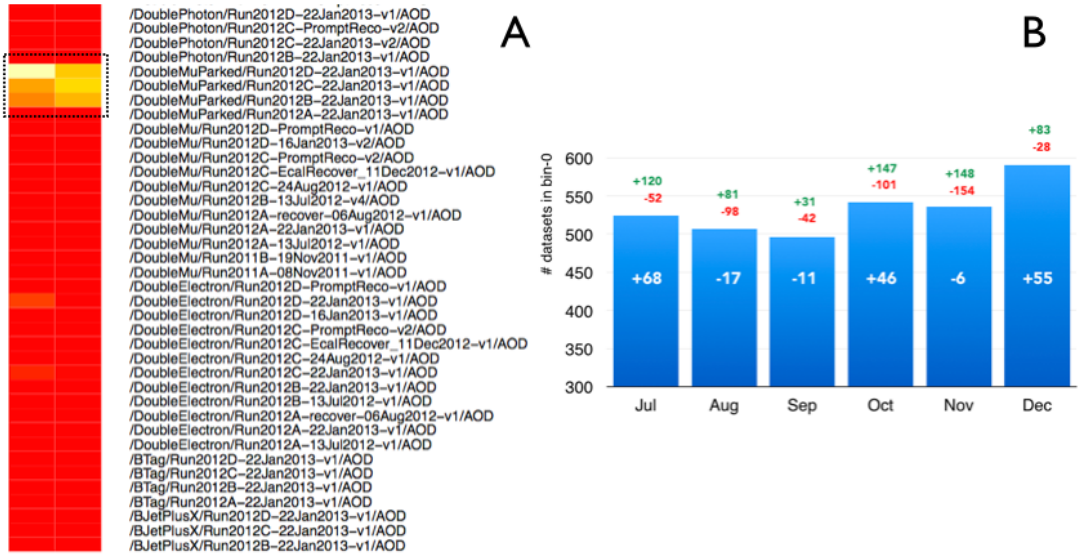

Figure 4. A. Example of a heat-map showing dataset-level correlation in "bin-0" (see text for explanation). B. Monthly increase (green) and decrease (red) in the number of non-accessed datasets at Tier-2 centers in second half of 2014 (see text for explanation).

to tell us more on what really is in the "bin-0". Deeper investigations on such bin-0 are possible, though. One option is to seek for correlation with specific classes of datasets, trying to identify those datasets who cause a Tier-2 disk occupancy that do not result in any data access for long time (e.g. parked datasets, see e.g. Figure 4A). Another option is to investigate more complex correlation, e.g. if the incremental increase/decrease of the number of non accessed datasets in the bin-0 (see e.g. Figure 4B) can be connected to the number of CMS contributions to conferences, thus quantitatively showing e.g. the recurrent changes in the data access patterns just before important conferences (preliminary studies on a partial data sample shows a tepid anti-correlation, about 30%, between the number of unpopular datasets and the number of CMS talks to conferences).

In a nutshell, the population of the bin-0 is one good example of an area that may benefit of deeper investigations aimed at seeking correlations with other observables: there is a chance to 'model' the dataset accesses from past data, in a way that allows to predict the access patterns in the near future.

5. A pilot framework for CMS analytics

A pilot project to understand the metrics, the analysis workflow and necessary tools (with possible technology choices) needed to attack such problems was desgined and implemented, and it is called DCAFPilot (Data and Computing Analysis Framework Pilot) [13]. The framework has been recently exploited to investigate the feasibility of this analytics approach for the use-case of the CMS data popularity described in the previous section.

The pilot architecture is graphically shown in Figure 2. First of all, data is collected from CMS data services by a DCAF core that relies on MongoDB for its internal cache. At the moment, structured data are collected from DBS, PhEDEx, PopDB, SiteDB, Dashboard (see [14] for more details on these CMS systems). In some more details: all datasets from DBS are collected into the internal cache; popular datasets are queried from the Popularity Database with a weekly granularity; for all of these datasets more information is also extracted from DBS,

PhEDEx, SiteDB and the Dashboard. This information is then complemented with random set of unpopular datasets (to avoid bias in machine learning algorithms used in a later stage).

Secondly, a data-frame generator toolkit collects and transforms data from these CMS data services and extract necessary bits for a subset of popular and un-popular datasets. This data-frame is then fed to machine learning algorithms for data analysis (both python and R code is used). Finally, a quantitative estimate of the popularity is given for specific types of datasets: this may eventually be fed back to the CMS computing infrastructure as a useful input to the CMS daily operations, as well as strategical choices.

At the current status, all data from 2013 and 2014 years have already been pre-processed and are available for analysis. And a prediction of which dataset(s) may become popular can be given, in the form of their probability versus each dataset name.

Before going into the results, some statistics from a dry run of the machinery are reported . Five data services were queried (for DBS, 4 instances were used) via 10 different APIs. The internal MongoDB-based cache was fed with about 220k datasets, 900+ release names, 500+ SiteDB entries, 5k people's DNs. About 800k queries were placed in total. Anonymization of potentially sensible information in these queries is done via the internal cache. The final data-frame coming out of this stage is constructed out of 78 variables, it is made out of 52 data-frame files, for a total of roughly 600k rows. Each file is worth about 1 week of CMS meta-data (in terms of size, it is approximately 600kB gzipped), and it corresponds to about 1k popular datasets with a roughly 1:10 ratio of popular vs unpopular samples randomly mixed. The data-frame can also be visualised real-time in terms of live data, correlations, as well as exploring different data popularity metrics (for example, number of users accessing a dataset versus total CPU used while accessing it).

The actual analysis can start only once the data collection is over. A data transformation step happens, and transforms the data into a suitable format for machine learning techniques. A specific machine learning approach must be chosen at this stage, e.g. either classification (allowing only a classification into categories, namely popular or unpopular) or regression (allowing to predict values for the metrics of choice, which may for example be the number of accesses) or online learning techniques. Currently, only a classification approach has been attempted. Then, the following step is to train and validate the machine learning model itself, which in this case has been done by splitting the data into train and validation sets (the 600K rows of the 2014 dataset have been organised to use the Jan-Nov sample as a train set, and the Dec sample as the validation set, i.e. the predictive power of the model is estimated on the validation set). At this moment several machine learning algorithms are adopted within DCAFPilot project: a regular set of scikit-learn classifiers [15], e.g. Random Forest, SGDClassifier, SVC, etc., the online learning algorhtm, Vowpal Wabbit [16], by Yahoo, and gradient boosting tree solution (xgboost, the eXtreme Gradient Boosting) [17]. Next stage is that new data are collected (e.g. early 2015) and they are transformed exactly as the 2014 dataset. The model chosen as the best one is finally applied to the new data to perform predictions, and such predictions must be regularly verified with fresh data from the Popularity Database once the metrics become available.

A major milestone within the DCAFPilot project has been completed, in terms of building up the machinery, which means collect data at regular intervals, transform them into a machine learning data-format, run various machine learning algorithms, yield predictions and compare them.

All the results have to be taken as preliminary, but they are encouraging. We are able to predict with reasonable accuracy a portion of the 2014 data set (see Table 1). None of the classifiers were especially tuned during this exercise: instead, we concentrated on the general approach and built all necessary tools to automate the procedure and build and maintain a stable pilot.

Once these results will be considered solid they will offer valuable information to tune CMS computing operations. For example, a few false-positive popularity predictions would imply some wasted bandwidth and disk space for a while, but not much at the end. On the other hand, a false-negative can mean a delay of a few days in a specific analysis. In one considers the experience of the Higgs announcement - in which data taken up to two weeks earlier was included in the analysis, with plots produced on that same morning - is teaching us that in hot periods a delay of few days could be important.

Classifier	naccess > 10			
	accu	prec	reca	F1
Random Forest [15]	0.98	0.86	0.98	0.92
SGDClassifier [15]	0.96	0.98	0.62	0.76
Linear SVC [15]	0.95	0.68	1.00	0.81
Vowpal Wabbit [16]	0.96	0.98	0.69	0.74
xgboost [17]	0.98	0.82	0.98	0.90

Table 1. Preliminary results from DCAFPilot. These predictions are performed by using nine months of 2014 for training, and predicting October data using different machine learning algorithms. The training set was split 66%/33% among training/validation and four statistics metrics were calculated, namely accuracy (accu), precision (prec), recall (reca) and F1-score.

6. Conclusions and next steps

Over years of operation in Run-1 and LS1 we collected large volumes of data that can be used for a better understanding of our Computing Model and for tuning our operations at the maximum efficiency. This data includes structured data sources (such as CMS data-services which hold information in relational databases) as well as unstructured data available via HyperNews, twikies, calendars, etc.

We launched a CMS Analytics project, with the ultimate goal to build adaptive models of CMS Computing. It is subdivided in sub-projects, each focussing on a specific use-case. In this paper, we showed a proof-of-concept based on the single use case of the CMS dataset popularity, and discussed its current status.

The DCAFPilot project has been used so far to build a working pilot to identify necessary data sources, machinery and tools. It has demonstrated to be capable of collecting the data, transforming them into a suitable format for machine learning, running various machine learning algorithms and making reasonable predictions. The final predictions can be verified a-posteriori by comparison with 'real' popularity data collected in the Popularity Database.

With such machinery now in place we are ready to start a full analysis. Our approach will be the following: collect historical data on a weekly basis, run them through all needed transformation and modelling steps, compare different classifiers and build the best predictive model and, finally, apply such model to a new set of data we expect to have.

Aside for the specifics of the data popularity use-case, this pilot gives us confidence that we can apply the same approach to other use-cases, thus adding more bricks towards the goal of achieving adaptive models of CMS computing.

References

[1] CMS Collaboration, "*The CMS experiment at the CERN LHC*", JINST **3** S08004 (2008)
[2] T. Barrass et al, "*Software agents in data and workflow management*", Proc. CHEP04, Interlaken, 2004. See also http://www.pa.org

[3] R. Egeland et al., "*Data transfer infrastructure for CMS data taking*", XIII International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT'08), Erice, Italy, Nov 3-7, 2008 - Proceedings of Science, PoS (ACAT08) **033** (2008)
[4] D. Spiga et al., "*The CMS Remote Analysis Builder (CRAB)*", Lect. Notes Comput. Sci. 4873 580-586 (2007)
[5] J. Andreeva et al., "*Distributed Computing Grid Experiences in CMS*", IEEE Trans. Nucl. Sci., vol. 52, p. 884-890, ISSN: 0018-9499
[6] J. D. Shiers, "*The Worldwide LHC Computing Grid (worldwide LCG)*", Computer Physics Communications 177 (2007) 219–223
[7] WLCG: http://lcg.web.cern.ch/lcg/
[8] D. Giordano et al, "*Implementing data placement strategies for the CMS experiment based on a popularity model*", J.Phys.Conf.Ser. 396 (2012) 032047
[9] C. Paus et al., "*Dynamic Data Management for the Distributed CMS Computing System*", this conference
[10] CMS Collaboration, "*The CMS Computing Project Technical Design Report*", CERN-LHCC-2005-023
[11] M. Aderholz et al., "*Models of Networked Analysis at Regional Centres for LHC Experiments (MONARC), Phase 2 Report*", CERN/LCB 2000-001 (2000)
[12] D. Bonacorsi, L. Giommi, V. Kuznetsov, T. Wildish, "*Exploring patterns and correlations in CMS Computing operations data with Big Data analytics techniques*", proceeding of International Symposium on Grids and Clouds (ISGC) 2015, March 2015, Academia Sinica, Taipei, Taiwan (submitted)
[13] https://github.com/dmwm/DMWMAnalytics/tree/master/Popularity/DCAFPilot
[14] M. Giffels, Y. Guo, V. Kuznetsov, N. Magini and T. Wildish, *The CMS Data Management System*, J. Phys.: Conf. Ser. 513 042052, 2014
[15] scikit: http://scikit-learn.org/stable/"
[16] Vowpal Wabbit: https://github.com/JohnLangford/vowpal_wabbit/wiki
[17] xgboost: https://github.com/dmlc/xgboost