PROCEEDINGS OF SCIENCE

Exploring patterns and correlations in CMS Computing operations data with Big Data analytics techniques

Valentin Kuznetsov
Cornell University
E-mail: vkuznet@gmail.com

Tony Wildish
Princeton University
E-mail: awildish@princeton.edu

Luca Giommi
University of Bologna, Italy
E-mail: luca.giommi2@studio.unibo.it

Daniele Bonacorsi*
University of Bologna and INFN-Bologna, Italy
E-mail: daniele.bonacorsi@unibo.it

The CMS experiment at the LHC accelerator at CERN designed and implemented a Computing model that allowed successful Computing operations in Run-1 (2009-2012) and gave a crucial contribution to the discovery of the Higgs boson by the ATLAS and CMS experiments. The workflow management and data management sectors of the model have been operated at full capacity exploiting WLCG resources for years. Around the massive volume of original and derived physics data from proton-proton and heavy-ions collisions in CMS, plenty of other data and metadata about the performances of the computing operations have been also collected and rarely (or never) examined. This latter sample is a wild mixture of non-physics heterogeneous data, both structured and unstructured, which well fits to deeper investigation with Big Data analytics approaches. In the context of CMS R&D activities, exploratory projects have been started to extract some values from this dataset and to seek for patterns, correlations as well as ways to simulate the Computing Model itself. Such studies will be presented and discussed.

International Symposium on Grids and Clouds (ISGC) 2015,
15 -20 March 2015
Academia Sinica, Taipei, Taiwan

*Speaker.

PoS(ISGC2015)008

1. Introduction

The CMS experiment [1] launched a Data Analytics project, whose goal is manyfold and depends on the timeline. As a long-term goal (2-3 years), the project aims to build adaptive data-driven models of CMS Data Management (DM) and Workload Management (WM) activities - as part of the overall CMS Computing Model [2] - with the target to be able to predict future behaviours of the CMS systems in operations from the detailed measurements of their performances in the past. As a medium-term goal (hopefully within LHC Run-2 (2015-2017) already, aiming for incremental improvements), the project aims to improve the use of CMS computing resources. As a short-term goal (within Run-2), the projects aim to concretely support the CMS Computing Operations team as much as possible through deeper understanding of the CMS data collected over the years. In this sense, understanding the "data" - by which we mean any (meta-)data produced by any Computing Operations activity since Run-1 started in 2009 - is extremely valuable in itself. The reason for such modelling to be adaptive is that models elaborated in the past aren't going to apply to the future for long, and only adaptive modelling itself will give CMS confidence and predictive power in the long term. The project is subdivided in a number of sub-projects. We add all possible ideas in a pool of potentially interesting projects, and work to make them self-contained and well-defined in scope and timeline. Only once a sub-project, with a limited amount of manpower investment in a short time, is found to be promising and useful for CMS Computing Operations, the sub-project is actually started and pursued until completion. The sub-projects hence start from specific details we want to learn more about in specific aspects of CMS workflows and of how our systems work, and a pilot project is launched only when aforementioned conditions are met. For the rest, we learn as we go.

2. Motivation and approach to data

During Run-I CMS built an operations model that worked and successfully met all requirements. It served well the CMS physics program, according to all possible metrics. But apart from Run-1 success we cannot claim that we fully understand our system and neither know how efficient we were during this time of operations. For example, the CMS Computing Model relies on a set of Grid Tier centres among which different kinds of data are exchanged, and in particular the data transfers to Tier-2 sites exceed our expectations as from the MONARC [3] model, moreover we have no model to shape this kind of traffic. Another example, we filled the disk storage at the Tier-2 level with plenty of data useful for physics analyses, but a large fraction of this data (their format name in the CMS jargon is AOD/AODSIM for data and Monte Carlo respectively) are stored and such storage is left un-accessed for long periods of time which accounts for inefficient resource utilisation. These are only a few examples which strongly suggest that deep understanding of all system components is required to improve its utilisation.

In the Computing operations during Run-1 and the first long shutdown LS1 (2013-2015) period, all computing systems collected plenty of data about operations themselves, e.g. monitoring data, accounting information, machine logs, etc. All of them were archived, but rarely (or never) accessed by anyone. This data comprises information about transfers, job submissions, site efficiencies, release details, infrastructure performance, analysis throughput, and much more. This

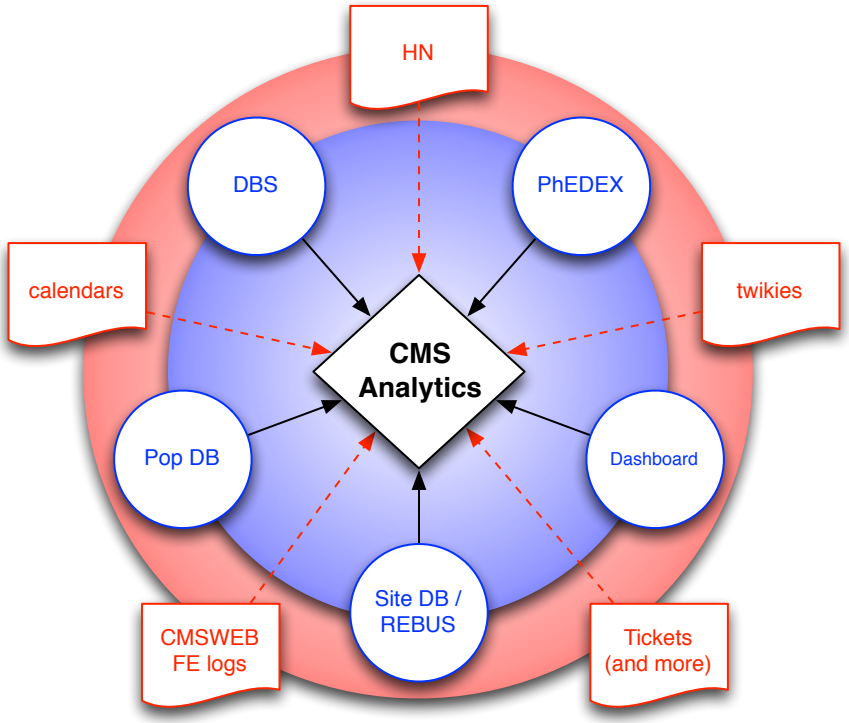

Figure 1: Source of structured and unstructured data to the CMS Analytics project. See text for explanations.

precious data set is left unanalysed so far because we mainly monitor our systems in near-time for debugging purposes, rather than analyse what happened in the past and study in depth systems behaviour. Additionally, we never fixed holes in our monitoring data and validated (most of) them with decent care. Therefore such data can be considered as incomplete and not suitable for further analysis. The quality of the data, and a careful work on data preparation before the analysis, is one of the main components which leads to success stories in any Big Data analytics project. In our case we must pay significant attention to the four big V's: the Volume (scale of the data), Velocity (analysis of streaming data), Variety (different forms of data), Veracity (uncertainty of data). The data Volume here is not negligible in itself, but definitely manageable with respect to the LHC collisions data we deal with. The Velocity is partially relevant, i.e. we aim to a quick availability of analytics results, but having a real-time feedback from the data is not actually a requirement. The Variety is very relevant, as we deal with a very irregular data set, consisting of structured, semi-structure and unstructured data (see next section). The Veracity is also extremely delicate, as the data integrity and the ability to trust the data analysis outcome to make decisions is crucial.

3. Structured data and beyond

Structured information (see Fig. 1, blue color) represents a variety of CMS Computing activ-

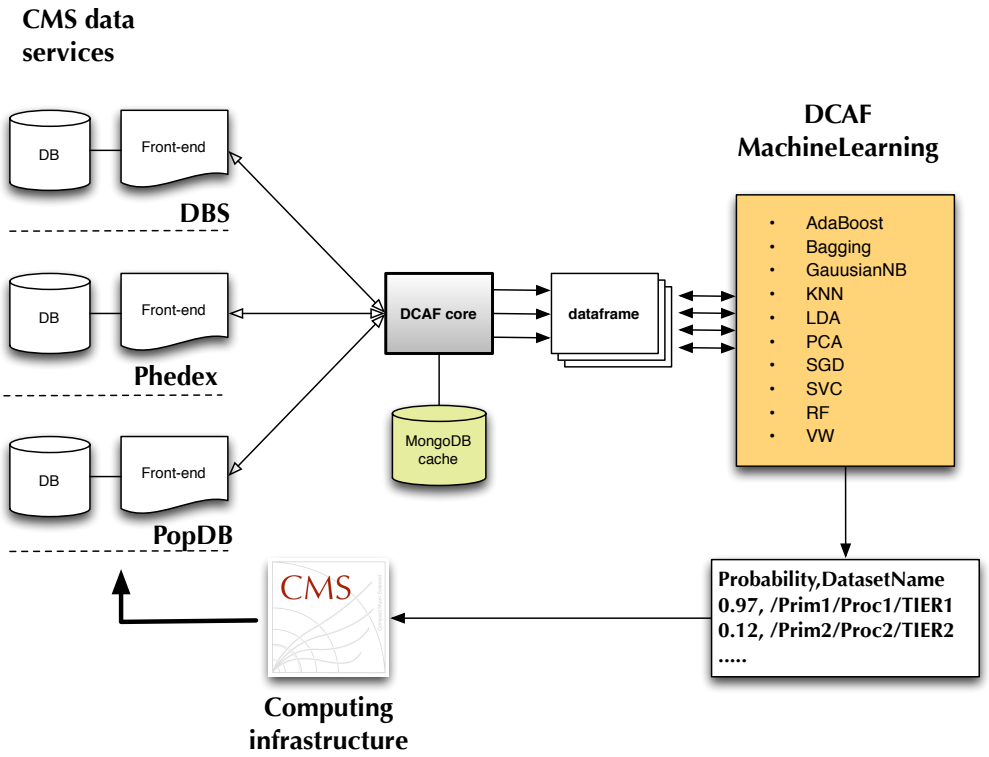

Figure 2: Descripton of the DCAFPilot workflow and components.

ities which are stored across multiple data services[1]. For example, the DBS system is the CMS source for physics meta-data; the PhEDEx transfer management database offers data transfer service and data replica catalog functionalities; the Popularity Database (PopDB) collects dataset user access information (e.g. access frequency, which replicas are accessed on Tiers of the WLCG [4, 5], the amount of CPU used); SiteDB (and REBUS, eventually) gives authoritative information about site pledges, deployed resources, and manpower onsite, while CERN Dashboard stands as a massive repository of details on Grid jobs (and beyond). In addition to structured data, plenty of data in the CMS Computing ecosystem is completely unstructured (see Fig. 1, red color). This type of information is hard to collect and process, but it represents potentially very rich content.

For instance, the CMS HyperNews system offers today more than 400 different fora, representing de-facto a reference on several years of user activities (announcements, information on user activities, insight into change of focus in the physics interests of individuals/groups over time, hot topics, etc.). But to be useful, it requires social data mining efforts on several aspects of collaboration-level activities.

The tickets offers a view on infrastructure issues reporting/tracking, via different tracking systems used over the years (e.g. Savannah [6], GGUS [7]), and complemented by activity-based electronic logbooks (ELOGs [8], topical e-groups [9], etc.)

[1] All information in CMS is available via CMS data service APIs.

The CERN-based TWikis offer a content that stands as a knowledge graph that could be mapped to user activities and physics interests, and help to model their evolution over the time.

Hot periods of CMS physics analysis can be tracked via the CMS calendar of events (as well as non-CMS calendars), the CMS sub-projects planning information, list of major conferences and workshops, etc. Such information can also lead to identification of seasonal cycles within different physics communities. To this extent, also the logs of the Vidyo [10] videoconference system may turn up to be useful, in terms of knowing who regularly attends specific meetings.

Finally, CMSWEB cluster front-end logs (actually, semi-structured data in this case) serve all data sources to users, thus might be mined to extract valuable information on user activities.

4. The CMS data popularity use-case on the CMS analytics pilot framework

As stated in the previous section, different sources of information are potentially useful to mine migration of physicists interests and evolution of their activities. If such data is properly handled and mined it can be a sensitive predictor of user activities within CMS collaboration which can be used to improve throughtput and efficiency of our computing system. The approach outlined so far was to identify several well formulated "problems" that may lead towards a solution. Under the umbrella of CMS Analytics project we try to investigate how to attach these problems by exploring different tools, technologies and various approaches via a set of pilot sub-projects. Below we'll discuss the dataset popularity pilot project in great details.

In CMS the Dynamic Data Placement [11] team is relying on historical information of datasets popularity to add (remove) replicas of existing datasets that appear to be most (least) desired by end-users. The goal of this activity is to balance resource utilisation at different sites by replicating more replicas of popular dataset. But this approach has one flaw; it reacts to spikes of the dataset

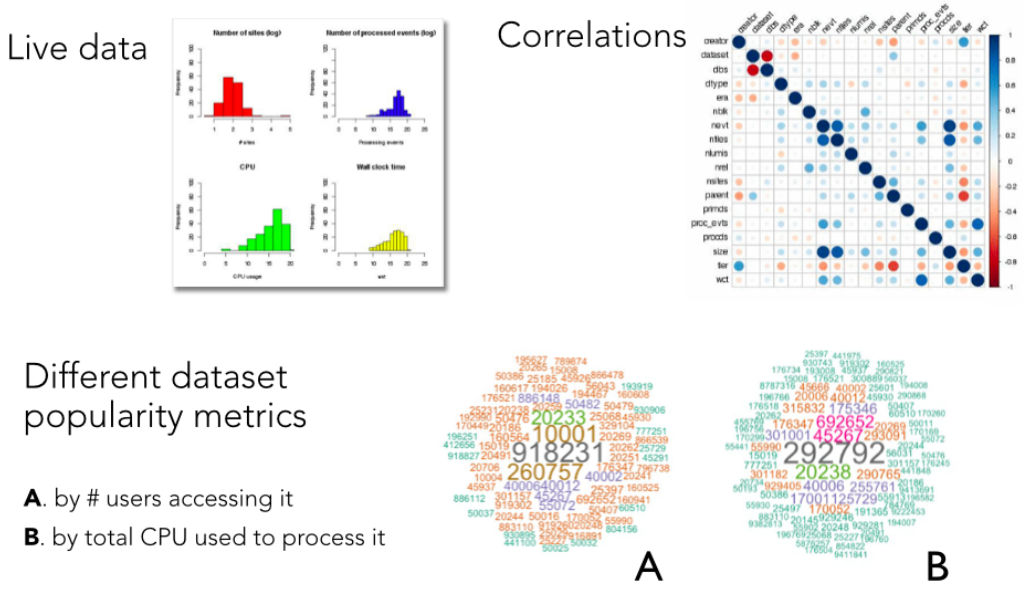

Figure 3: Graphical composition of some ways to visualise the data frame (see text for explanation).

popularity after the fact. Therefore it would be desired to have a system in place which can predict which datasets will become popular even before datasets will be available on Grids for further analysis.

The DCAFPilot (Data and Computing Analysis Framework Pilot) [12] is a pilot project to understand the metrics, the analysis workflow and necessary tools (with possible technology choices) needed to attack this problem. The framework has been recently exploited to investigate the feasibility of this analytics approach for the CMS data popularity use-case.

In a nutshell, the pilot architecture is shown in Figure 2. Data is collected from CMS data services[2] by a DCAF core that uses MongoDB for its internal cache. A data-frame generator toolkit has been developed to collect and transform data from CMS data services, and to extract necessary bits for a subset of popular and un-popular datasets. The data-frame is fed to machine learning algorithms (both Python and R code used) for data analysis. A quantitative estimate of the popularity is given for specific types of datasets, which may be fed back to the CMS computing infrastructure as a useful input to daily operations and strategical choices.

The data collection flows, in some more details, works as follows. All datasets from DBS are collected into the internal cache. Popular datasets are queried from PopDB with a weekly granularity. For all of these datasets more information is also extracted from DBS, PhEDEx, SiteDB and the Dashboard. This information is complemented with random set of unpopular datasets (to avoid bias in later stage machine learning algorithms). All such information is stored in different data-frame files, that can be fed to any machine learning library for whatever purpose one may have. At the moment, all data from 2013 and 2014 years have already been pre-processed and are available for analysis. At this stage, a prediction of which dataset(s) may become popular is given, in the form of their probability versus each dataset name.

Some statistics from a dry run of the machinery are reported below. Five data services were queried (4 DBS instances used) and 10 APIs were used. The internal MongoDB cache was fed with about 220k datasets names, 900+ CMS software release names, 500+ SiteDB entries, 5k people's Distinguished Names (DNs). In total, about 800k queries were placed. Anonymisation of potentially sensible information is done via the internal cache. The final data-frame is constructed out of 78 variables, and made out of 52 data-frame files, roughly 600k row in total. Each file is worth 1 week of CMS meta-data (approximately 600kB gzipped), and it has about 1k popular datasets with a roughly 1:10 ratio of popular vs unpopular samples randomly mixed. The data-frame can be visualised real-time (see Fig. 3) in terms of live data, correlations, and also exploring different data popularity metrics (e.g. number of users accessing a dataset versus total CPU used to process it).

Once the data collection is finalised, the actual analysis can start. First of all, a data transformation is needed to transform the data into a suitable format for machine learning techniques. Then, a specific machine learning approach must be chosen, e.g. classification (it allows only to classify into categories, e.g. popular or unpopular) versus regression (it allows to predict real values of the chosen metrics, e.g. number of accesses) vs online learning techniques (so far a classification approach has been adopted). The following step is to training and validate the machine learning

[2] So far, we collect information from the following CMS structured data-services: DBS, PhEDEx, PopDB, SiteDB, Dashboard: see [13] for more details on the CMS systems.

model. This is done by splitting the data into training and validation sets. The 600K rows in the 2014 dataset has been organised so to use the January-November sample as a train set, and the December sample as the validation set, i.e. the predictive power of the model is estimated on the validation set. Then, 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 then applied to such new data to make predictions, and such predictions are regularly verified with PopDB fresh data once metrics become available.

Within the DCAFPilot project we have completed a major milestone to build-up the machinery, i.e. collect data on regular intervals, transform them into machine learning data-format, run various machine learning algorithms and yield and compare predictions. At this moment several machine learning algorithms are adopted within DCAFPilot project: a regular set of scikit-learn classifiers [14], e.g. Random Forest, SGDClassifier, SVC, etc., the online learning algorhtm, Vowpal Wabbit [15], by Yahoo, and gradient boosting tree solution (xgboost, the eXtreme Gradient Boosting) [16].

The preliminary results are quite encouraging. For instance we are able to predict with reasonable accuracy a portion of the 2014 data set (see Table 1). We used the first nine months of data for training, and predicted the October data using a different set of classifiers. 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.

All the results have to be taken as preliminary, but they already give interesting indications, and once they will be considered solid they will offer valuable information to tune CMS computing operations. A few examples may clarify how. A few false-positive popularity predictions would imply a little wasted bandwidth and disk space for a while, but not much. On the other hand, a false-negative can mean a few days delay in a specific analysis. The experience of the Higgs announcement - in which data taken two weeks earlier was included, with plots produced 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 [14]	0.98	0.86	0.98	0.92
SGDClassifier [14]	0.96	0.98	0.62	0.76
Linear SVC [14]	0.95	0.68	1.00	0.81
Vowpal Wabbit [15]	0.96	0.98	0.69	0.74
xgboost [16]	0.98	0.82	0.98	0.90

Table 1: Preliminary results from the DCAFPilot package. To make these predictions we used nine months of 2014 for training and predicted october data using different machine learning algorithms. The traininig set was split 66%/33% as training/validation sets and four statistics metrics were calculated: accuracy (accu), precision (prec), recall (reca) and F1-score.

So far the DCAFPilot project is capable of collecting the data, transforming them into machine learning suitable format, run various machine learning algorithms and make a predictions. The final predictions can be verified posteriory by comparing them with data collected in PopDB. With such machinery in place we're ready to start full analysis. Our approach is the following: collect historical data on weekly basis, run them through transformation and modeling steps, compare

different classifiers and built best predictive model, and, finally apply this model to new set of data we expect to have. The latter can be collected from Request Manager and DBS CMS data-system by the time data-ops team start processing the datasets.

5. Conclusions

Over many years of operation in Run-1 we collected large amounts of data which can be used to fine tune our Computing Model. This information includes, but is not limited to, structured data sources (such as CMS data-services which hold information in relational databases) as well as unstructured data available via HyperNews, TWikis, calendar and other sources. We are able to identify the main directions of the CMS Analytics project and break it down into well defined sub-projects. The latter can be used as pilots to identify necessary sources, machinery and tools to successfully build adaptive models of CMS Computing. In this paper, we show a proof-of-concept based on the single use case of the CMS dataset popularity, and discussed its current status. The DCAFPilot project demonstrated that we can successfully collect, build and train machine learning algorithms with CMS computing data and make reasonable predictions. This gives us a useful tool to evaluate different approaches in data collection, mining and prediction as well as moving towards the goal of achieving an adaptive model of CMS computing via CMS Analytics.

References

[1] CMS Collaboration, "*The CMS experiment at the CERN LHC*", JINST **3** S08004 (2008)

[2] CMS Collaboration, "*The CMS Computing Project Technical Design Report*", CERN-LHCC-2005-023

[3] M. Aderholz et al., "*Models of Networked Analysis at Regional Centres for LHC Experiments (MONARC), Phase 2 Report*", CERN/LCB 2000-001 (2000)

[4] J. D. Shiers, "*The Worldwide LHC Computing Grid (worldwide LCG)*", Computer Physics Communications 177 (2007) 219–223

[5] WLCG: http://lcg.web.cern.ch/lcg/

[6] http://savannah.web.cern.ch/savannah/

[7] https://ggus.eu/

[8] https://midas.psi.ch/elog/

[9] https://e-groups.cern.ch/

[10] http://www.vidyo.com/

[11] https://twiki.cern.ch/twiki/bin/viewauth/CMS/DynData

[12] https://github.com/dmwm/DMWMAnalytics/tree/master/Popularity/DCAFPilot

[13] M. Giffels, Y. Guo, V. Kuznetsov, N. Magini and T. Wildish, *The CMS Data Management System*, J. Phys.: Conf. Ser. 513 042052, 2014

[14] scikit: http://scikit-learn.org/stable/"

[15] Vowpal Wabbit: https://github.com/JohnLangford/vowpal_wabbit/wiki

[16] xgboost: https://github.com/dmlc/xgboost