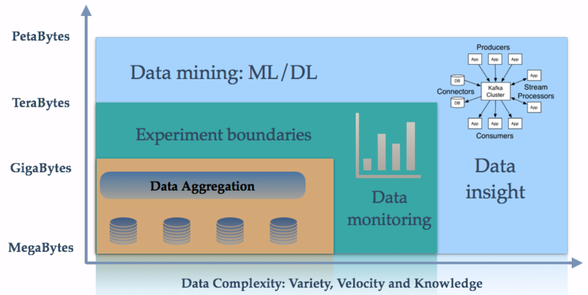
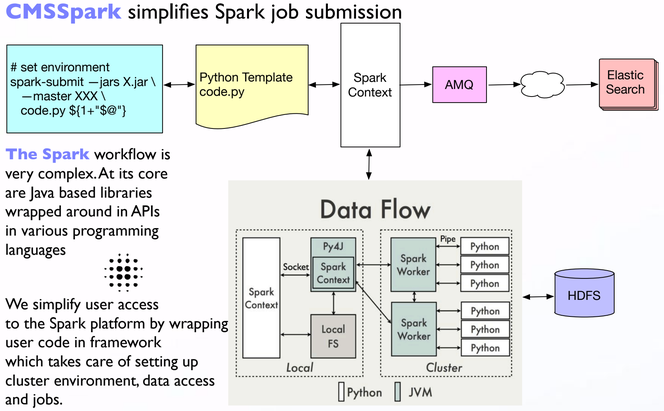
Gaining Insight From Large Data Volumes with Ease

Valentin Kuznetsov, Cornell University



We see **an exponential** acceleration of data growths in HEP field. Only few years ago the HEP was dominated by RDBMS data services. In recent years the traditional methods of hosting experiment meta-data were complemented by NoSQL databases (MongoDB, CouchDB, Cassandra), Hadoop+Spark cluster computing frameworks, user-friendly notebooks (IPython, Swan, etc.) as well as HDFS file system.

CMSSpark simplifies Spark job submission

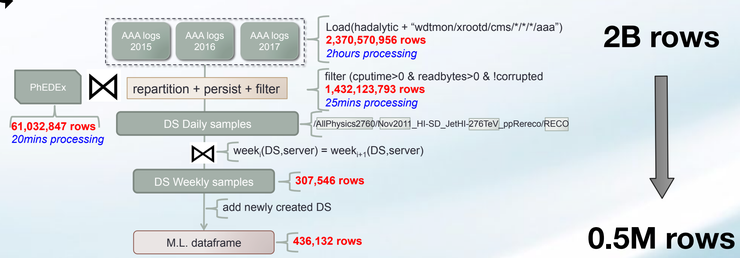


The Spark workflow is very complex. At its core are Java based libraries wrapped around in APIs in various programming languages

We simplify user access to the Spark platform by wrapping user code in framework which takes care of setting up cluster environment, data access and jobs.

run_spark	dbs_agg.py	–yarn –date 2018 –fout hfs:///cms/users/vk
shell wrapper	user workflow	user options

Spark provides platform for data pre-processing for ML tasks



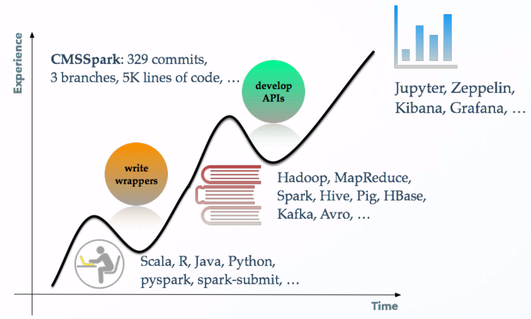
2B rows

0.5M rows

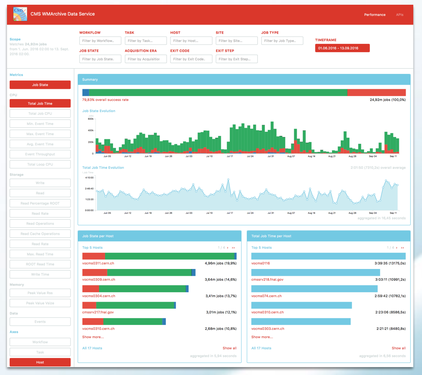
Machine Learning starts playing a significant role in daily CMS operations. We apply it not only for physics analysis but also to better understand our computing resources. For example, we use ML studies for better data placement, monitoring and anomaly detections. Quite often the data frame for ML are pre-processed on HDFS via Spark framework.

We use **Hadoop Spark** platform for data reduction and processing. Jobs take hours to run on 40 nodes cluster and we reduce data by factor of 5000.

Nowdays users need to know broad variety of tools, from different programming languages to understanding how to run complex workflows in clustering environment. Quite often users experience **steep learning** curve. This includes but is not limited to databases, notebooks, data-formats, processing tools, algorithms, visualization techniques and Machine Learning.



We successfully deployed **WMArchive data-service** to store and aggregate CMS WMAgent Framework Job Reports. The system collects 100M+ docs per year and provides aggregated information about CMS data-processing. **HDFS** and **Spark** plays critical role in a system



On-going and future ML projects in CMS

Job-scheduling
perform studies of 200M+ FWJR records from WMArchive in a context of processing time, memory utilization and turned around into ML problem for allocating resources of future jobs on a grid nodes. This may become important studies for using opportunistic resources.

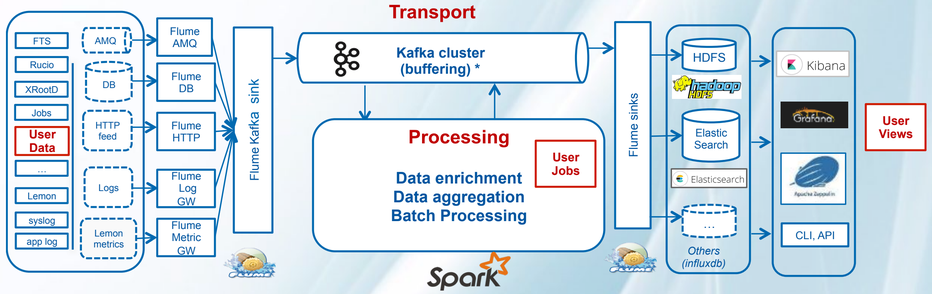
Workflow Estimate Time of Completion (ETC) from images
use monitoring images to estimate workflow time of completion. Use HTCondor graphs from gwmsmon and workflow time of completion from ReqMgr2. Predict workflow ETC using Deep Convolutional Network (regression vs multi-class categorization).

Job clustering
use WMAgent, CRAB FWJR documents, and HTCondor logs to perform studying user activities (clustering)

Network optimization
study network security by identifying anomalies in network traffic; predicting network congestion; dynamic routing; bug detection via analysis of self-learning networks, and WAN path optimization based on user access patterns.

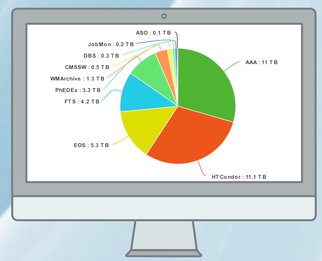
AutoEncoder for data compression
use AutoEncoder to find better representation of our data, e.g. NANOAOD

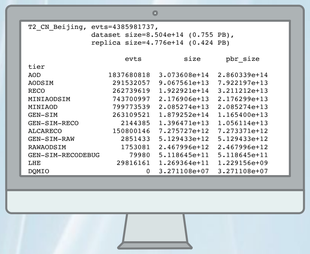
TensorFlow as a Service (TFaaS)
development of common service to serve various ML models for CMS users



CERN MONIT system consists of
100+ data producers, 3.5 TB/day
3 days of data retention in Kafka
13 Spark jobs 24/7

CERN Analytix cluster:
39 nodes with 64GB of RAM, 32 cores/node
Mix of Intel® Xeon® CPU E5-2650 @ 2GHz
AMD Opteron(™) 6276

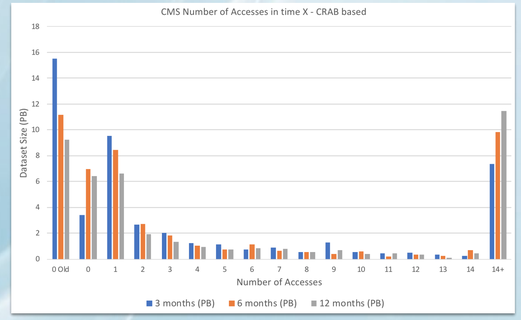




We provide various dashboards as well as final datasets for our users. Data are then used for further analysis. The Spark platform significantly improved our capabilities to gain additional insight from our data.

CMS data on HDFS contains more than 32 TB in total and is stored in various data-formats (CSV, JSON, Avro). In need to deal with dozens of data streams coming from various CMS sources. The following represents user activities.

In dedicated dashboards we perform **cross-check analysis** and discuss obtained results at different venues.



CMS scrutiny plot was produced by combining HTCondor, DBS, PhEDEx data and processing them on Spark platform.