CMS RPC Non-Physics Event Data Automation

Anton Dimitrov
University of Sofia
on behalf of CMS collaboration

RPC2024, XVII Conference on RPC Chambers and Related Detectors
9-13.Sep.2024, Santiago de Compostela, Spain

History

The RPC non-physics event (a.k.a condition) data automation project started back in 2016 to boost:

- the RPC Current Taskforce initiated in June 2014 to investigate the RPC detector current evolution and determine possible reasons for the worrisome tendency in current increase
- the aging studies by developing an automate to integrate the RPC currents from the FWCAENCHANNEL table in the CMS_RPC_PVSS_COND database schema.

Later it was further developed to different studies of current dependancy on various parameters like instaneous luminosity and environmental data.

Finally, in 2023-2024 the paradigm was changed from hardware channel (DPID) to ChamberID and many new automates were created to populate new DB tables with chamber driven data.

New ML methods are applied for RPC current prediction

Initial Motivations

1. Deeply study the RPC current dependence on different physical and environmental parameters such as LHC luminosity, gas channel flow, UXC relative humidity, UXC temperature, background radiation, HV board offset fluctuations etc…
2. Clean up ramp up spikes in stable monitored voltage – big problem at that time on WBM tool for all DOC,DM,PFB shifters.
3. RPC aging studies are extremely important for future projection of detector performance

    i. RPC Integrated Charge Studies

    ii. RPC Detector Performance Studies

Problems

There were actually two main problems:

A. The amount of information stored in the FWCAENCHANNEL table was simply impossible to work with just using direct interaction of SQL with the database.
B. The information in this table was not synchronous, it was change based, i.e. only one parameter (Imon, Vmon, Status) was stored at a change for a given timestamp, all the other DPE are void.

A + B ➔ extremely heavy query which never finishes and even crashes the server.

203826		8863		29-OCT-15 06.20.59.175000000 AM
203826		8681		29-OCT-15 06.21.09.126000000 AM
203826		8500		29-OCT-15 06.21.19.181000000 AM
203826		8340		29-OCT-15 06.21.29.189000000 AM
203826		8158		29-OCT-15 06.21.39.190000000 AM
203826		7976		29-OCT-15 06.21.49.184000000 AM
203826		7794		29-OCT-15 06.21.59.268000000 AM
203826		7612		29-OCT-15 06.22.09.256000000 AM
203826		7431		29-OCT-15 06.22.19.235000000 AM
203826		7271		29-OCT-15 06.22.29.245000000 AM
203826		7089		29-OCT-15 06.22.39.276000000 AM
203826		6907		29-OCT-15 06.22.49.226000000 AM
203826		6748		29-OCT-15 06.22.59.269000000 AM
203826		6566		29-OCT-15 06.23.09.246000000 AM
203826			1	29-OCT-15 06.23.19.231000000 AM
203826		6500		29-OCT-15 06.23.19.233000000 AM
203826	.3000000119			29-OCT-15 06.23.39.189000000 AM
203826	.200000003			29-OCT-15 07.23.47.091000000 AM
203826		6499		29-OCT-15 08.24.06.457000000 AM
203826	.3000000119			29-OCT-15 09.24.49.924000000 AM
203826				29-OCT-15 11.07.52.302000000 AM
203826	7.5			29-OCT-15 11.08.12.248000000 AM
203826			3	29-OCT-15 11.08.12.251000000 AM
203826		7260		29-OCT-15 11.08.12.253000000 AM
203826		7484		29-OCT-15 11.08.22.423000000 AM
203826	7.800000191			29-OCT-15 11.08.32.535000000 AM
203826		7770		29-OCT-15 11.08.32.539000000 AM
203826		8026		29-OCT-15 11.08.42.449000000 AM
203826	8.199999809			29-OCT-15 11.08.52.495000000 AM
203826		8281		29-OCT-15 11.08.52.499000000 AM
203826	8.5			29-OCT-15 11.09.02.499000000 AM
203826		8504		29-OCT-15 11.09.02.504000000 AM
203826	8.800000191			29-OCT-15 11.09.12.553000000 AM
203826		8759		29-OCT-15 11.09.12.559000000 AM
203826	9.100000381			29-OCT-15 11.09.22.543000000 AM
203826		9047		29-OCT-15 11.09.22.548000000 AM
203826		9270		29-OCT-15 11.09.32.470000000 AM
203826	1.899999976			29-OCT-15 11.09.42.512000000 AM
203826			1	29-OCT-15 11.09.42.515000000 AM
203826		9345		29-OCT-15 11.09.42.517000000 AM
203826	1.5			29-OCT-15 11.09.52.547000000 AM
203826	1.100000024			29-OCT-15 11.15.12.744000000 AM
203826	.8999999762			29-OCT-15 12.15.42.595000000 PM
203826				29-OCT-15 12.20.52.528000000 PM
203826	0			29-OCT-15 12.21.22.547000000 PM
203826			5	29-OCT-15 12.21.22.550000000 PM
203826		8790		29-OCT-15 12.21.22.553000000 PM
203826		8608		29-OCT-15 12.21.32.470000000 PM
203826		8426		29-OCT-15 12.21.42.517000000 PM
203826		8267		29-OCT-15 12.21.52.492000000 PM
203826		8085		29-OCT-15 12.22.02.509000000 PM
203826		7880		29-OCT-15 12.22.12.569000000 PM
203826		7721		29-OCT-15 12.22.22.669000000 PM
203826		7539		29-OCT-15 12.22.32.607000000 PM
203826		7357		29-OCT-15 12.22.42.662000000 PM
203826		7198		29-OCT-15 12.22.52.775000000 PM
203826		7016		29-OCT-15 12.23.02.758000000 PM
203826		6811		29-OCT-15 12.23.12.825000000 PM
203826		6674		29-OCT-15 12.23.22.740000000 PM
203826			1	29-OCT-15 12.23.32.788000000 PM
203826		6500		29-OCT-15 12.23.32.790000000 PM
203826	.3000000119			29-OCT-15 12.39.23.342000000 PM
203826		6499		29-OCT-15 01.40.37.100000000 PM
203826	.200000003			29-OCT-15 02.41.05.297000000 PM
203826	.3000000119			29-OCT-15 05.41.54.339000000 PM
203826	.200000003			29-OCT-15 06.41.57.649000000 PM
203826	.3000000119			29-OCT-15 07.42.05.822000000 PM

Elapsed: 00:00:00.02

Solution

Process information in very small chunks. Such that the database doesn't have any problems with the amount of information.

We decided to attack the problem by requesting the same query many many many times but with a small window (We request 1 day at a time the information). This basically translates to a lot of executions.

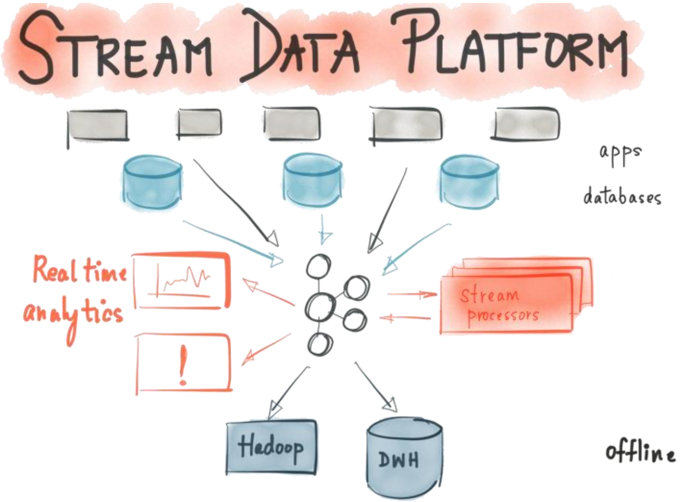
But for the database the density of the requests is not an issue, since it's designed to be a high demand service.

➔ Define a method “Current Probe” with a very low CPU load, used as a backbone in the entire automation.

Automation Features: Data Streaming

Working with big data it's very common to use streaming techniques because you only run over the information once and never store in memory or files large amounts of data.

The idea is to keep the flow moving.



Automation Features: Multithreading

The automat would have to be a multithreaded algorithm because it would have to run for many parameters which could potentially take a lot of time. So you can spawn many threads to address multiple requests at the same time

Requests
a.k.a
Feed

(Input)

Concurrent
Request
Dispatcher

(CRD)

Worker 1
(Thread 1)

Worker 2
(Thread 2)

Worker 3
(Thread 3)

Worker N
(Thread N)

IBERO
CIUDAD DE MÉXICO ®

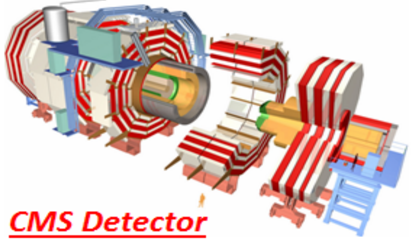
Universidad de
los Andes

Evolved Motivations

- Data Synchronization: synchronize the raw data to obtain values for all important parameters like Vmon, Imon, Status for every single timestamp.
- **Data Tagging**: implement binary flag to tag every single current with HV status data (like Off, Offset, Standby, RPC ON) and combine with data from external tables like LHC luminosity and CMS Magnetic Field.
- Create a complete automation to retrieve, synchronize and analyse the raw data on the fly and to reorder and store the newly analysed data into different database tables defined to cover various RPC Current Related Tasks like
    - Integrated Charge
    - Current dependence on LHC Inst. Luminosity
    - Current dependence on Environmental Parameters
    - Current Evolution in time
    - HV Conditioning
    - HV Conditioning Fit
    - Dynamically define RPC SG/DG OM
    - Offset Subtraction

Realization

DCS – PVSS/WinCC OA



CMS Detector

ConfDB

Development Environment

Archiving ~19000 DPNAMES

- Power system channels (FWCAENCHANNELS): Current, Voltage, etc...
- Gas Channels & gas system parameters
- Environmental parameters: T, P, RH

CondDB

DB cms_omds_lb → 1:1 → DB cms_omds_adg

schema CMS_RPC_PVSS_COND → Automation → schema CMS_RPC_COND

Asynchronous *FWCAENCHANNEL* table in CMS_RPC_PVSS_COND

DPID		NUMBER
CHANGE_DATE		TIMESTAMP(9)
DPE_STATUS		NUMBER
DPE_POSITION		NUMBER
READBACKSETTINGS_RDWN		NUMBER
ACTUAL_IMON		NUMBER
READBACKSETTINGS_I0		NUMBER
ACTUAL_VMON		NUMBER
READBACKSETTINGS_RUP		NUMBER
READBACKSETTINGS_V0		NUMBER
ACTUAL_STATUS		NUMBER
READBACKSETTINGS_TRIPTIME		NUMBER
UPDATEID	NOT NULL	NUMBER(38)
SETTINGS_I0		NUMBER
SETTINGS_V0		NUMBER
ACTUAL_TEMP		NUMBER
ACTUAL_TRIP		NUMBER
ACTUAL_VCON		NUMBER

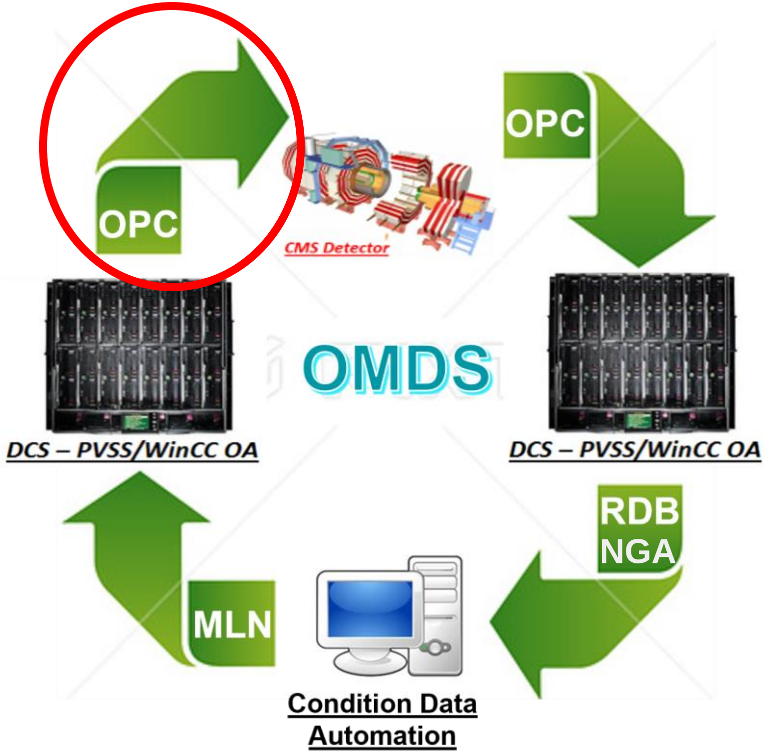
New synchronous *RPCCURRENTS* table in CMS_RPC_COND

RPCCURRENTS

DPID	NOT NULL	NUMBER
CHANGE_DATE		TIMESTAMP(9)
IMON		NUMBER
VMON		NUMBER
STATUS		NUMBER
HV FLAG		NUMBER
LHC FLAG		NUMBER

Ideology: Full Circle

- 5 Main Automates synchronizing raw data
  - ✓ UXC Environment (PVSS data)
  - ✓ RPC Gas Channel Flow (PVSS data)
  - ✓ LHC Blocks (PVSS data)
  - ✓ RPC Currents (PVSS data)
  - ✓ RPC Rates (root files from /rpctdata)
- RPC Current Studies with relational parameters
- RPC Rates Studies
- Virtual Objects: region, wheel/disk, station/ring, sector - Objects can be conveniently defined to study the behaviour along **Z, R, ϕ** in the CMS coordinate system
- Change of paradigm: chamber-centric architecture
- ML models for Current prediction



Main Features: Data Synchronization, Data Tagging, Relational Studies, Complete Automation, Full Circle

RPC Data Automation

RPC Automation → RPC DB Schemas → Development (CMS_RPC_PVSS_TEST @ INT2R)
Production (CMS_RPC_COND @ CMS_OMDS_LB)

RPC Automation → External Schemas → CMS_RUNTIME_LOGGER, CMS_WBM, CMS_DCS_ENV_PVSS_COND

CMS_OMS

SCAL (Run1,2) → TCDS (Run3)

LHCLog	UXC Environment
RPCCurrents	
Ramp Up Smoothing	Gasflow
Ramp Down Smoothing	
Offset	
Evolution	
Integrated Charge	
HVCOND	
HVCOND fitting	
Lumi fitting	

There are 21 automates in the RPC automation : one for each task related to the retrieving and analysing condition data.

These automates query the database, precook the data and store them in a database for future users to take without having to go through the “hell” of analysing raw data. This gives future analysis a very easy approach which in some cases is even solved by simple SQL queries to the precooked data.

Main Automates: UXC ENVIRONMENT

UXC Environmental parameters like Pressure, Temperature, Dewpoint and Relative Humidity synchronized for every single timestamp.

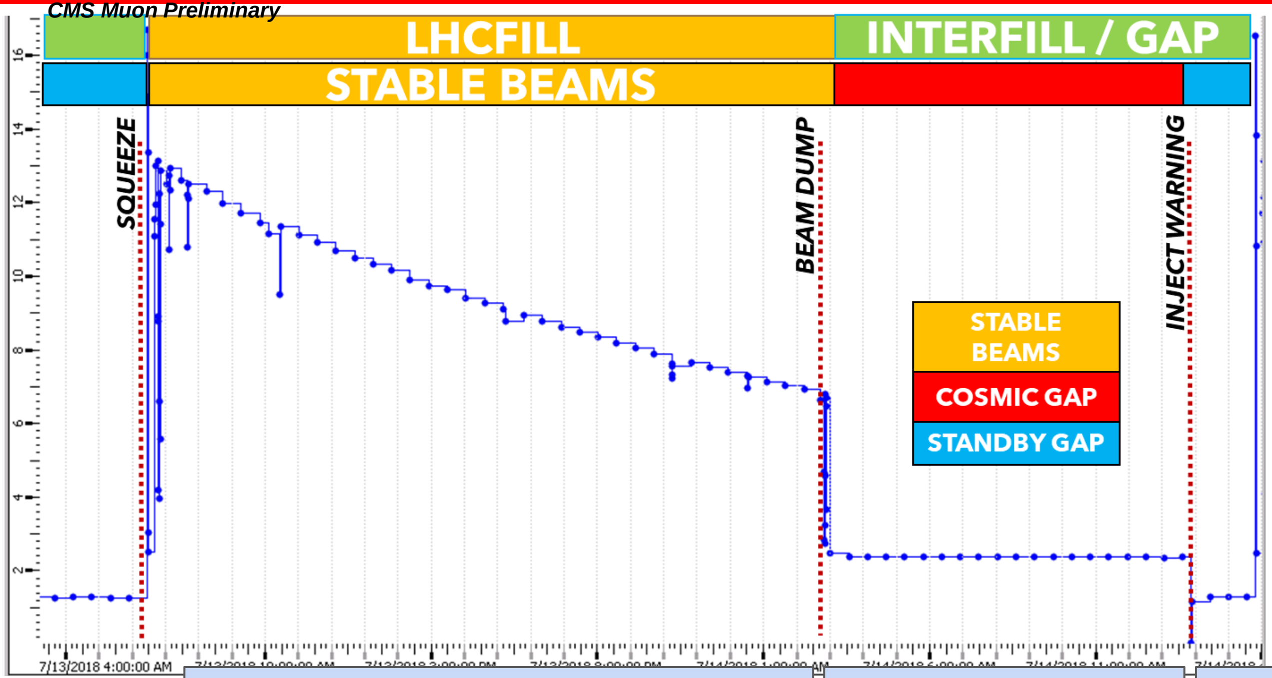
CHANGE_DATE	PRESSURE	TEMPERATURE	RELATIVE_HUMIDITY	DEWPOINT
01-JAN-14 02.37.05.000000000 AM	964.62	19.94	26.5	2.22
01-JAN-14 04.37.06.000000000 AM	964.03	19.94	26.5	2.22
01-JAN-14 06.37.07.000000000 AM	963.74	19.94	26.5	2.22
01-JAN-14 08.37.08.000000000 AM	963.52	19.94	26.5	2.22
01-JAN-14 10.37.08.000000000 AM	962.2	19.94	26.5	2.22
01-JAN-14 12.37.09.000000000 PM	960.66	19.94	26.5	2.22
01-JAN-14 02.37.10.000000000 PM	959.71	19.94	26.5	2.22
01-JAN-14 04.37.11.000000000 PM	958.97	19.94	26.5	2.22
01-JAN-14 06.37.11.000000000 PM	958.02	19.94	26.5	2.22
01-JAN-14 08.37.12.000000000 PM	957	19.94	26.5	2.22
01-JAN-14 10.37.13.000000000 PM	956.41	19.94	26.5	2.22
02-JAN-14 12.37.13.000000000 AM	955.38	19.94	26.5	2.22
02-JAN-14 04.37.14.000000000 AM	954.14	19.94	26.5	2.22
02-JAN-14 06.37.16.000000000 AM	955.38	19.94	26.5	2.22
02-JAN-14 08.37.16.000000000 AM	956.63	19.94	26.5	2.22
02-JAN-14 10.37.17.000000000 AM	957.58	19.94	26.5	2.22
02-JAN-14 12.37.18.000000000 PM	958.24	19.94	26.5	2.22
02-JAN-14 02.37.18.000000000 PM	958.46	19.94	26.5	2.22
02-JAN-14 04.37.20.000000000 PM	958.83	19.94	26.5	2.22

Main Automates: RPC GAS CHANNEL FLOW

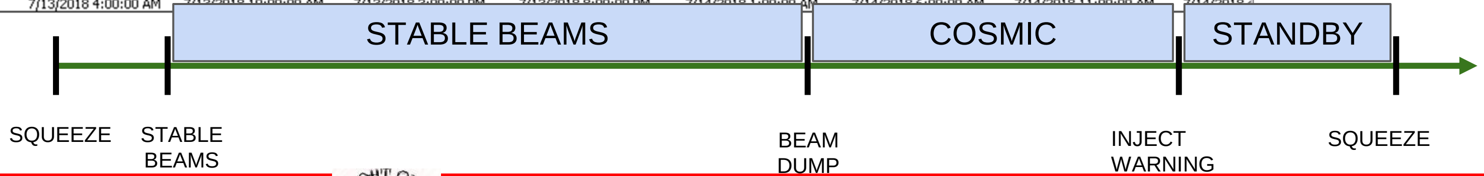
All 468 RPC gas flow cell flowIn and flowOut values have been synchronized to have both values for a single timestamp.

DPID	CHANGE_DATE	FLOWIN	FLOWOUT
16310	25-JUN-18 03.37.40.000000000 PM	31.6	30.7
16310	25-JUN-18 10.03.54.000000000 AM	31.5	30.3
16310	25-JUN-18 04.29.55.000000000 AM	31.5	30.5
16310	24-JUN-18 10.56.05.000000000 PM	31.5	30.7
16310	24-JUN-18 05.22.15.000000000 PM	31.3	30
16310	24-JUN-18 11.48.26.000000000 AM	31.4	30.4
16310	24-JUN-18 06.14.31.000000000 AM	31.4	30.2
16310	24-JUN-18 12.41.05.000000000 AM	31.4	30.4
16310	23-JUN-18 07.07.18.000000000 PM	31.7	30.5
16310	23-JUN-18 01.28.08.000000000 PM	31.6	30.2
16310	23-JUN-18 07.54.15.000000000 AM	31.6	30.3
16310	23-JUN-18 02.20.40.000000000 AM	31.4	30.2

Main Automates: LHCLOG BLOCKS



- During **stable beams** we perform current vs luminosity study and integrate collision currents to get the integrated charge in collisions.
- During **cosmic gap** we average cosmic currents for every single HV channel to monitor their evolution in time and integrate cosmic currents to get the integrated charge outside collisions.
- During **standby gap** we average standby currents for every single HV channel to monitor their evolution in time



RPC Currents: Data Synchronization & Tagging

Asynchronous *FWCAENCHANNEL* table in CMS_RPC_PVSS_COND

DPID	CHANGE_DATE	VMON	IMON	STATUS
315	25-JUN-17 11.58.29.083000000 PM	6501		
315	25-JUN-17 11.59.31.745000000 PM	6498		
315	26-JUN-17 12.03.46.240000000 AM		12.89999962	
315	26-JUN-17 12.03.46.240000000 AM			3
315	26-JUN-17 12.03.46.241000000 AM	8114		
315	26-JUN-17 12.03.49.194000000 AM	8240		
315	26-JUN-17 12.03.59.618000000 AM	8463		
315	26-JUN-17 12.04.03.358000000 AM	8592		
315	26-JUN-17 12.04.12.933000000 AM	8815		
315	26-JUN-17 12.04.16.631000000 AM	8878		
315	26-JUN-17 12.04.26.126000000 AM		13.30000019	
315	26-JUN-17 12.04.26.127000000 AM	9133		
315	26-JUN-17 12.04.29.838000000 AM		12.10000038	
315	26-JUN-17 12.04.29.838000000 AM	9226		
315	26-JUN-17 12.04.29.838000000 AM			1
315	26-JUN-17 12.04.39.462000000 AM		1	
315	26-JUN-17 12.04.39.463000000 AM	9231		
315	26-JUN-17 12.07.22.387000000 AM		0.699999988	

New synchronous *RPCCURRENTS* table in CMS_RPC_COND

DPID	CHANGE_DATE	VMON	IMON	STATUS	FLAG	TIME_DELTA
315	25-JUN-17 11.58.29.000000000 PM	6501	0.5	1	20	62
315	25-JUN-17 11.59.31.000000000 PM	6498	0.5	1	20	255
315	26-JUN-17 12.03.46.000000000 AM	8114	12.9	3	16	3
315	26-JUN-17 12.03.49.000000000 AM	8240	12.9	3	16	10
315	26-JUN-17 12.03.59.000000000 AM	8463	12.9	3	16	4
315	26-JUN-17 12.04.03.000000000 AM	8592	12.9	3	16	9
315	26-JUN-17 12.04.12.000000000 AM	8815	12.9	3	16	4
315	26-JUN-17 12.04.16.000000000 AM	8878	12.9	3	16	10
315	26-JUN-17 12.04.26.000000000 AM	9133	13.3	3	16	13
315	26-JUN-17 12.04.39.000000000 AM	9231	1	1	24	163
315	26-JUN-17 12.07.22.000000000 AM	9231	0.7	1	24	26

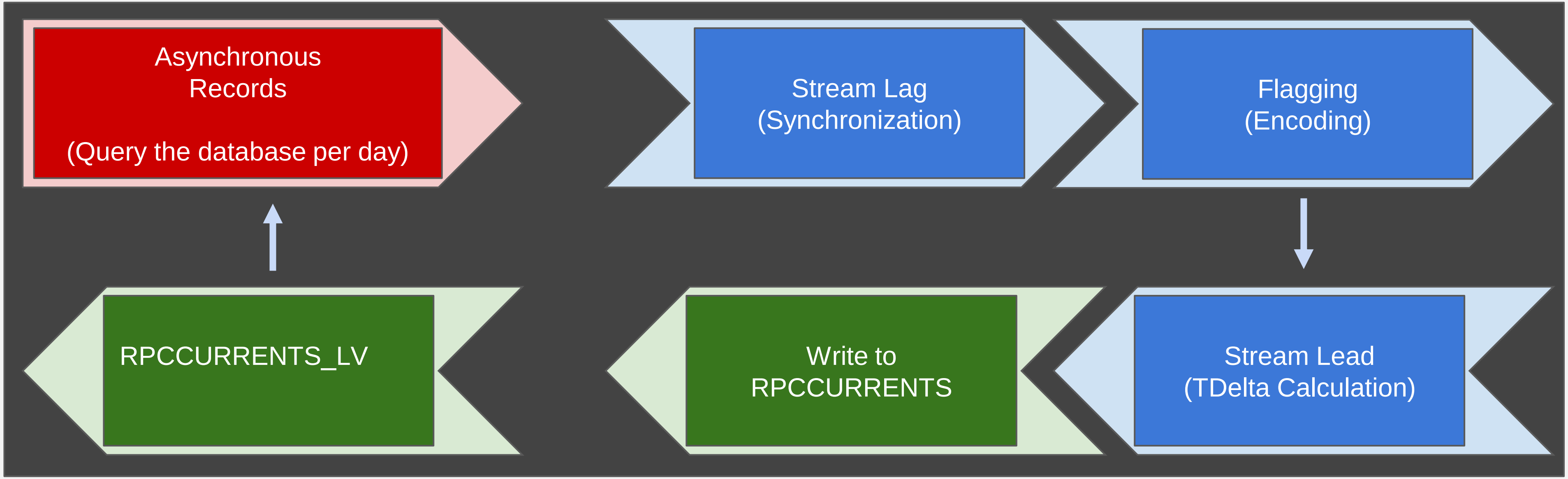
LUMI	BFIELD	RPC ON	STDB	OFFSET	OFF
32	16	8	4	2	1

- Data tagging based on
  - RPC HV stable condition
  - CMS Magnetic Field (CMS_DCS_ENV_PVSS_COND.CMSFWMAGNET)
  - LHC Luminosity (CMS_OMS.LUMISECTIONS)

RPCCurrents Population

Color Scheme

Red: Oracle, SQL
Blue: Multithreading, Streaming Libraries
Green: Oracle, DML



Example of RPCCurrents Automat

The Flagging and TDelta Calculation stages in the data flow are highly important given that they will reduce the stress on the database when being consulted by other tools (WBM, offline analysis, web applications).

RPC Currents: Ramp Up Smoothing

Problem : high ramp up currents with stable voltage and CAEN actual_status=1
Mechanism : the STATUS changes from 3 (RAMP UP) to 1 (STABLE) soon after the monitored voltage Vmon is close to the set applied voltage V0set.
Reason : due to the big capacitance of the RPC detector, when the monitored voltage Vmon is already stable, the current Imon still continues to decay (charge redistributes on the bakelite surface), so for couple of seconds (~10-ish) there appear between 3 to 5 Imon values which are much above the "stable" chamber operation current.
Solution : the values highlighted in yellow, in the example below, are reassigned to the last RAMP UP value with STATUS=3. To avoid a gap in time, the period that each of the data points holds is summed (15 seconds highlighted in yellow in the example below) and added to the last ramp up TIME_DELTA value with STATUS=3

DPID	CHANGE_DATE	VMON	IMON	STATUS	FLAG	TIME_DELTA
315	24-NOV-09 05.39.31.000000000 PM	833	18	3	0	1
315	24-NOV-09 05.39.32.000000000 PM	865	18	3	0	2
315	24-NOV-09 05.39.34.000000000 PM	898	18	3	0	3
315	24-NOV-09 05.39.38.000000000 PM	997	9.899999619	1	2	2
315	24-NOV-09 05.39.40.000000000 PM	997	5.599999905	1	2	4
315	24-NOV-09 05.39.44.000000000 PM	997	1.799999952	1	2	9
315	24-NOV-09 05.39.53.000000000 PM	997	1.399999976	1	2	354
315	24-NOV-09 05.45.47.000000000 PM	997	0.899999976	1	2	4
315	24-NOV-09 05.45.51.000000000 PM	997	1.200000048	1	2	49

DPID	CHANGE_DATE	VMON	IMON	STATUS	FLAG	TIME_DELTA
315	24-NOV-09 05.39.31.000000000 PM	833	18	3	0	1
315	24-NOV-09 05.39.32.000000000 PM	865	18	3	0	2
315	24-NOV-09 05.39.34.000000000 PM	898	18	3	0	18
315	24-NOV-09 05.39.53.000000000 PM	997	1.399999976	1	2	354
315	24-NOV-09 05.45.47.000000000 PM	997	0.899999976	1	2	4
315	24-NOV-09 05.45.51.000000000 PM	997	1.200000048	1	2	49
315	24-NOV-09 05.46.40.000000000 PM	997	1.600000024	1	2	3
315	24-NOV-09 05.46.43.000000000 PM	997	1.100000024	1	2	208
315	24-NOV-09 05.50.11.000000000 PM	997	1.5	1	2	19

RPC Currents: Ramp Down Smoothing

Problem : when the RPC detector is switched off, in the process of voltage ramp down at lower voltages, the ramp down STATUS=5 changes to 1 (STABLE) before going finally to 0 (OFF). It is a noise in the database, but most importantly, it can create confusion in the algorithms of other automates (i.e. Offset).

Solution : remove that noise without creating a gap in time, see example below

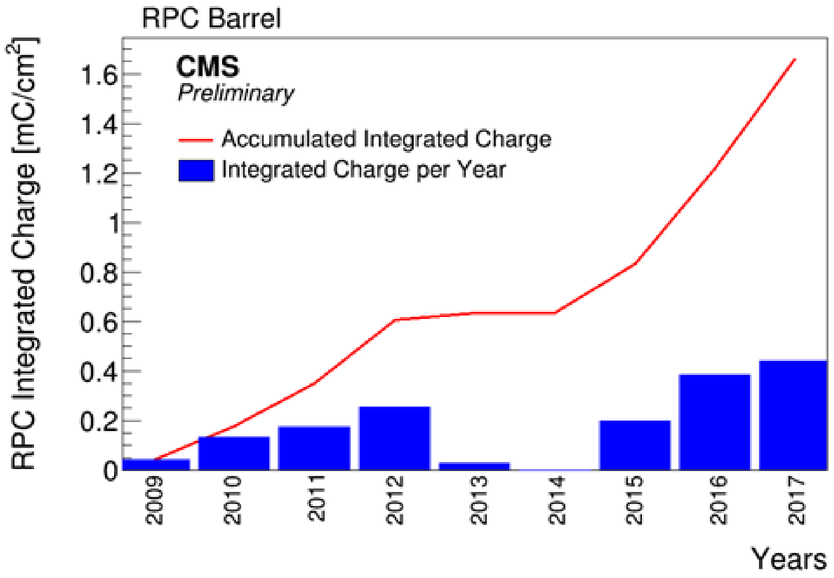
DPID	CHANGE_DATE	VMON	IMON	STATUS	FLAG	TIME_DELTA
315	01-OCT-09 06.36.42.000000000 PM	1287	0	5	0	4
315	01-OCT-09 06.36.46.000000000 PM	1274	0	5	0	4
315	01-OCT-09 06.36.50.000000000 PM	1263	0	5	0	2
315	01-OCT-09 06.36.52.000000000 PM	1252	0	5	0	4
315	01-OCT-09 06.36.56.000000000 PM	1240	0	1	0	2
315	01-OCT-09 06.36.58.000000000 PM	1240	0	0	1	2
315	01-OCT-09 06.37.00.000000000 PM	1228	0	0	1	4
315	01-OCT-09 06.37.04.000000000 PM	1216	0	0	1	4

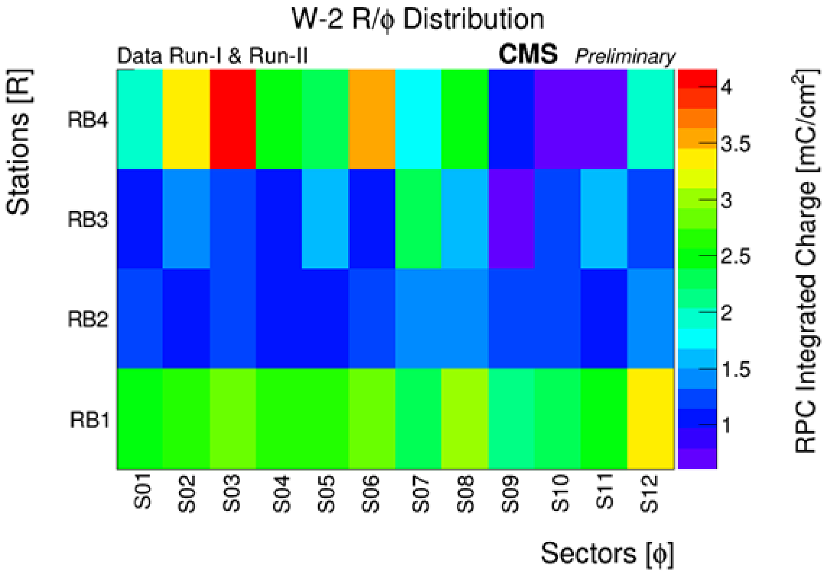
DPID	CHANGE_DATE	VMON	IMON	STATUS	FLAG	TIME_DELTA
315	01-OCT-09 06.36.42.000000000 PM	1287	0	5	0	4
315	01-OCT-09 06.36.46.000000000 PM	1274	0	5	0	4
315	01-OCT-09 06.36.50.000000000 PM	1263	0	5	0	2
315	01-OCT-09 06.36.52.000000000 PM	1252	0	5	0	6
315	01-OCT-09 06.36.58.000000000 PM	1240	0	0	1	2
315	01-OCT-09 06.37.00.000000000 PM	1228	0	0	1	4
315	01-OCT-09 06.37.04.000000000 PM	1216	0	0	1	4
315	01-OCT-09 06.37.08.000000000 PM	1197	0	0	1	6

RPC Current Study: Integrated Charge

RPC Integrated charge per single HV channel and Virtual objects is being calculated and a new daily value is stored in dedicated tables at the end of each day.

Additional automate is available to provide the **accumulated integrated charge** for virtual objects in 2D maps and the calculated extrapolations to accumulated integrated charge of 3000 fb^{-1}

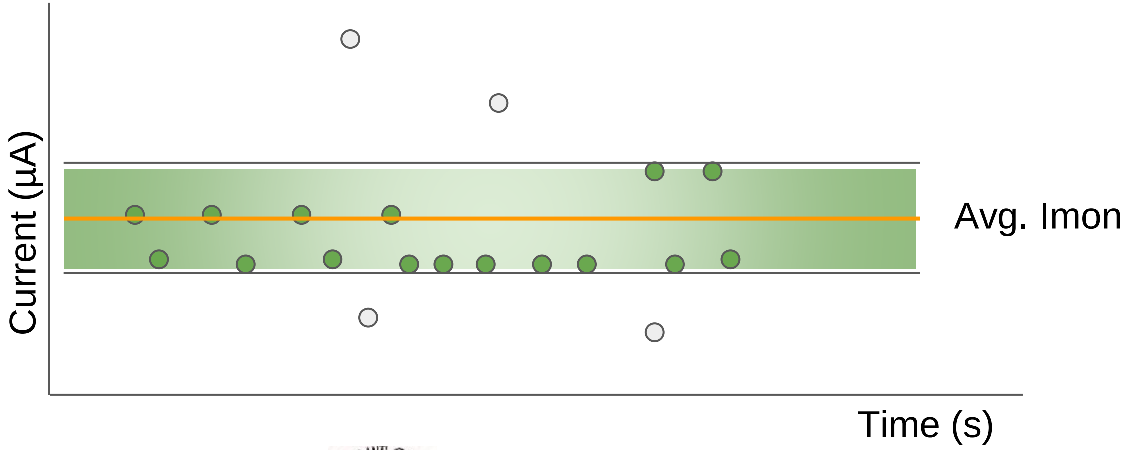




RPC Current Study: HV Conditioning

The **RPC HV Conditioning** is a current scan taken 3-4 times per year at the voltage steps:
1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 8500, 9000, 9100, 9200, 9300, 9400, 9500, 9600, 9700, 9800

Every single HV point is in fact a block of current data at a predefined, stable voltage. For every single data block a dedicated algorithm is applied to better calculate the average for the block. Finally, all blocks are stitched into a single set following the predefined HV points.



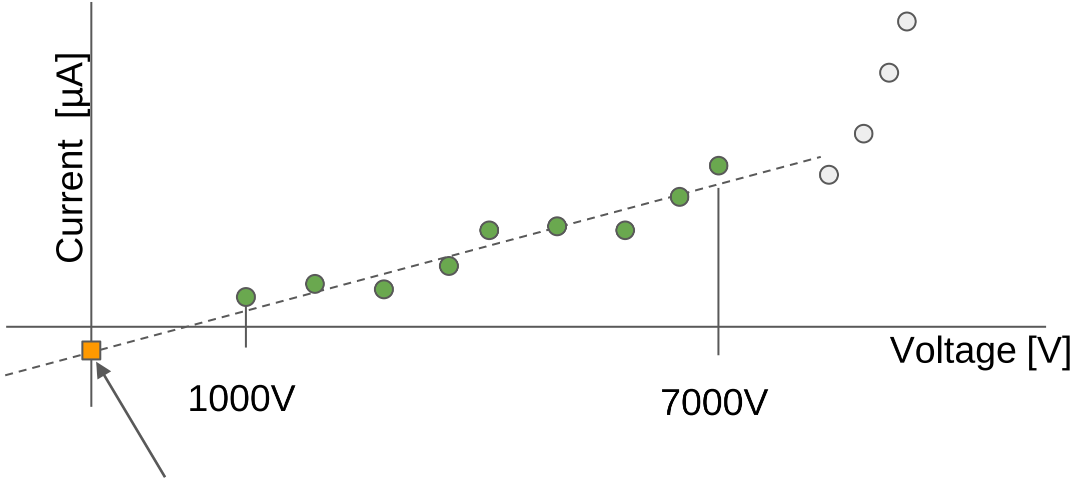
- We keep the points from the Imon distribution within +/- 4σ and remove anything outside the green area.
- We apply the same procedure to the remaining points iteratively till max 10% of initial points are cut.
- We take the mean as Avg. block current

IBERO
CIUDAD DE MÉXICO ®

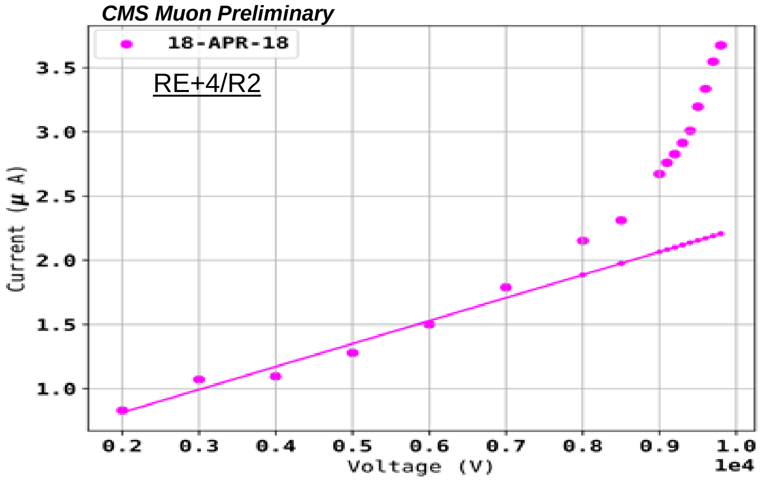
Universidad de los Andes

RPC Current Study: HV Conditioning Fit

Finally, the points between 1000 and 7000 V are linearly fit "on the fly" in the data streaming process to provide the fit parameters: **p1**, the reverse of the **Resistance**, and **p0**, the **HV channel offset.**



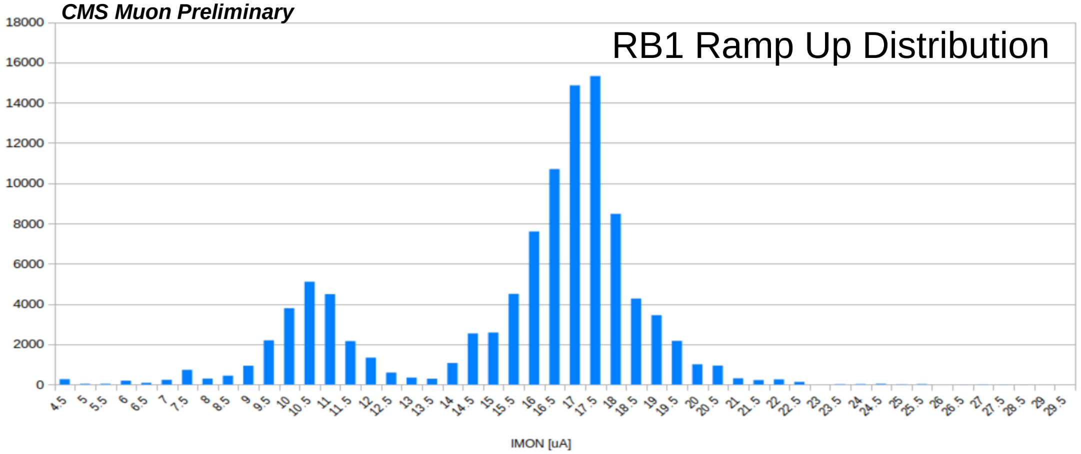
P0 = HV channel offset (it could be negative)

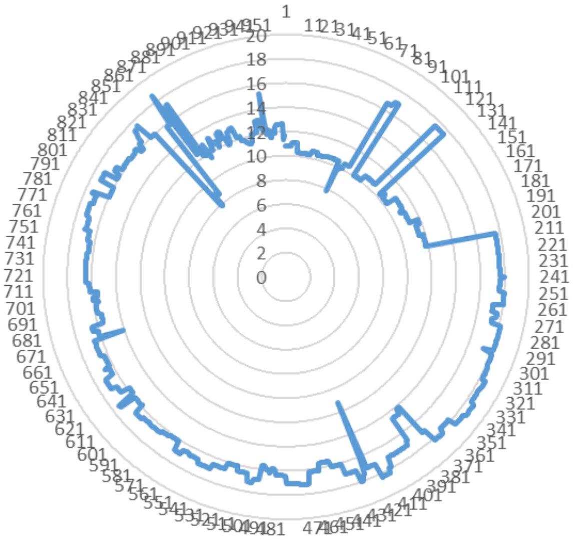


P0 and P1 plots in time are available in OMS to monitor the stability of the physical parameters behind the fits.

RPC Current Study: Dynamically Define OM

Looking at the ramp up currents between 500 - 1000V, dynamically define the RPC SG/DG OM as well as the timestamp of the OM flip ⇒ create change log per HV DPID.



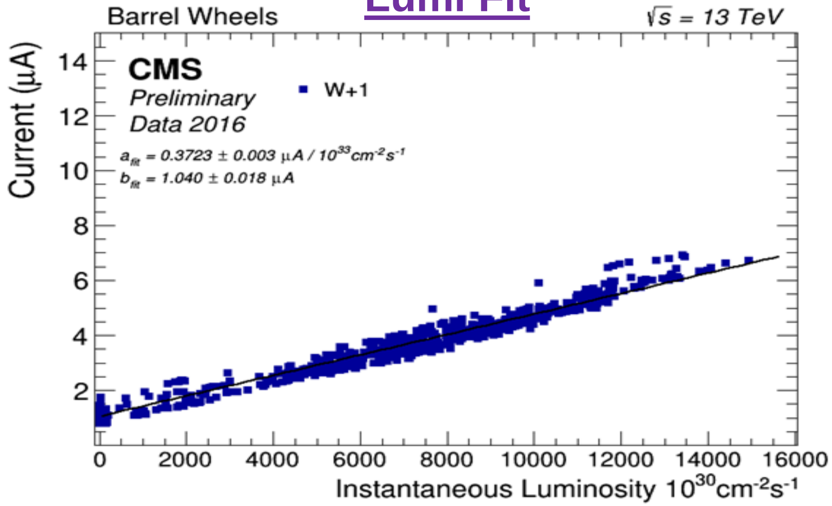


RPC Current Study: Lumi Fit & Evolution

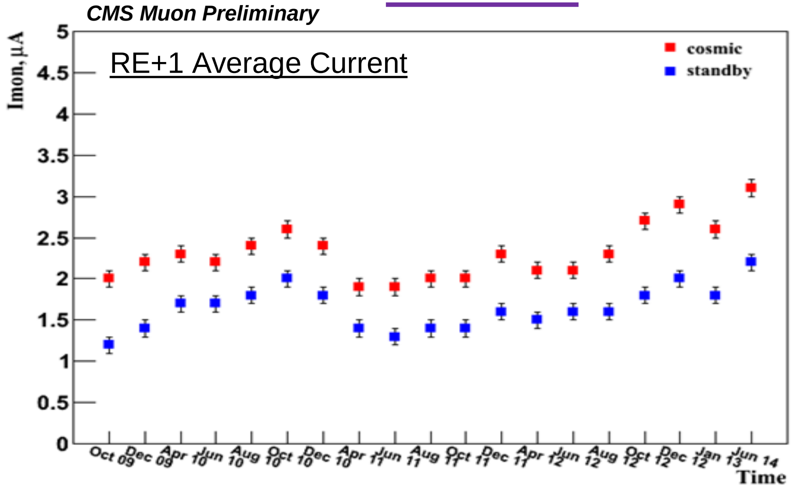
Using the information retrieved from the RPCCURRENTS table, we can analyse the current behaviour as a function of the instantaneous luminosity per fill (left) and the cosmic and standby current evolution in time (right). RPC Evolution automate finds and stores all HV channel trips, overcurrent spikes, initial ramps and offsets.

FILL	GAP	
FILL	COSMIC GAP	STANDBY GAP

Lumi Fit



Evolution

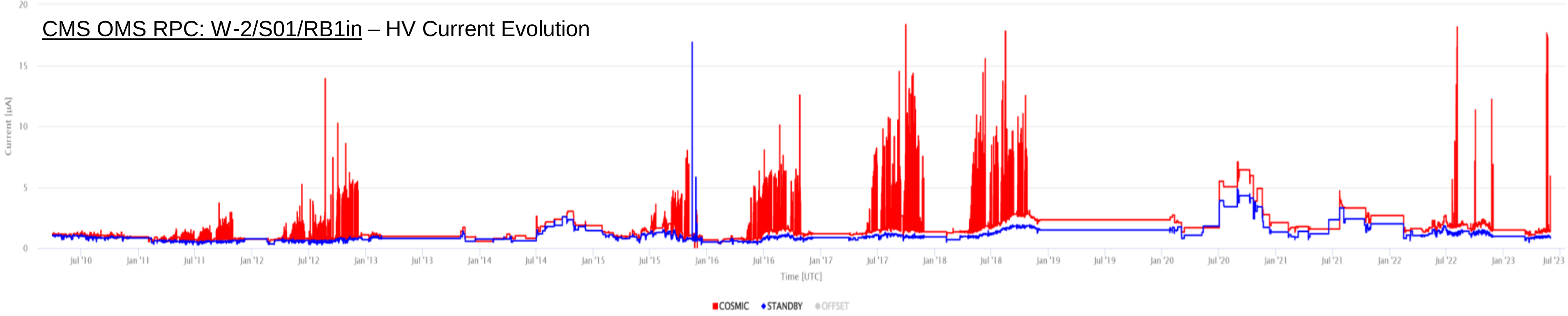


RPC Current Study: Evolution Problem 1

- Problem: Cosmic and Standby spikes in current evolution.
  - ✓ Collision currents are flagged as cosmic when unstable beams (levelling) ➔ cosmic spikes.
  - ✓ High currents during beam injection ➔ standby spikes.
- Solution: Match current blocks with LHCLOG blocks.
  - ✓ Match cosmic current blocks with LHC cosmic gaps.
  - ✓ Match standby current blocks with LHC standby gaps.

CMS Muon Preliminary

CMS OMS RPC: W-2/S01/RB1in – HV Current Evolution

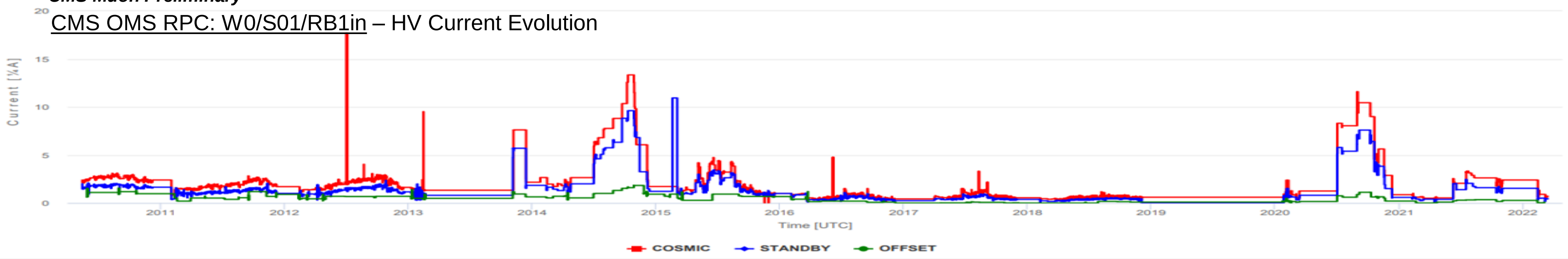


RPC Current Study: Evolution Problem 2

- Goal: Fast data chamber current plotting on OMS for the entire history (since beginning of Run-I) as shown below.
- Hardware: In Barrel 1 HV channel (DPID) powers 1 chamber while in Endcap 1 channel powers two chambers.
- Problem: Changing DPID mapping in time due to HV channel reshuffling does not allow plotting data fast:
  - ✓ Complicated implementation of HV channel to chamber mapping historically in time.
  - ✓ Heavy SQL query for fast plotting on OMS.
  - ✓ Long data retrieval
- Solution:
  - Perform heavy queries in an automate ➔ **change of paradigm**
  - Store data per chamber.

CMS Muon Preliminary

CMS OMS RPC: W0/S01/RB1in – HV Current Evolution



Change of Paradigm

- New paradigm:
  Switch from DPID to Chamber_ID.
  - ✓ We study the evolution of 1056 chambers instead of 781 DPID.
  - ✓ Data in database are stored per DPID & chamber IDs.
- During operation:
  - Record in database cosmic, standby, offset currents in the evolution table as TYPE.
  - Fast full period current plotting on OMS.

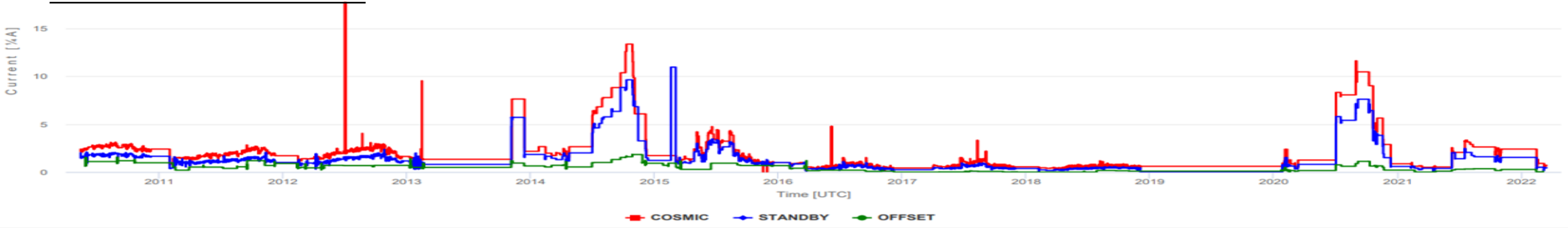
DPID	STARTTIME	STOPTIME	IMON	IMON_ERR	TYPE
...	...	...	...	...	...

From DPID to ChamberID

DPID	CHAMBER_ID1	CHAMBER_ID2	STARTTIME	STOPTIME	IMON	IMON_ERR	TYPE
...	...	...	...	...	...	...	...

CMS Muon Preliminary

CMS OMS RPC: W0/S01/RB1in – HV Current Evolution



Change of Paradigm: Accumulated Int. Charge

- Accumulated integrated charge corresponds to the total cumulative charge since the beginning of operation (01-OCT-09) until present partitioned daily.

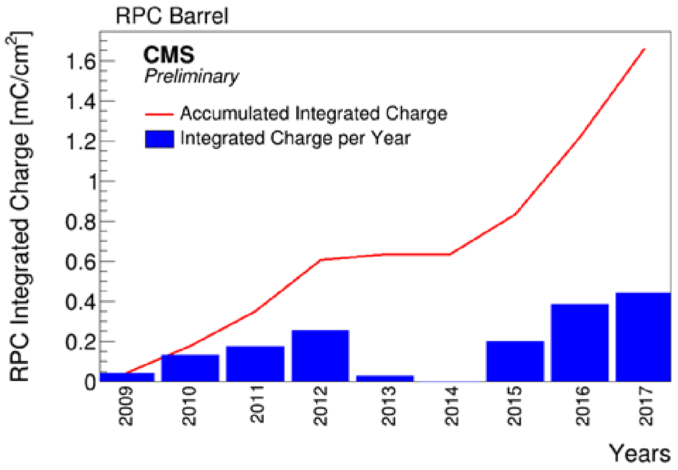
  - ✓ Two associated currents type have been defined (COLLISION and RPC_ON):

  - Accumulated integrated charge per DPID.

DPID	FROM	TO	ACC_INTEGRATED_CHARGE	TYPE
...	...	...	...	...

  - Accumulated integrated charge per chamber ID (New paradigm)

CHAMBER_ID	FROM	TO	ACC_INTEGRATED_CHARGE	TYPE
...	...	...	...	...

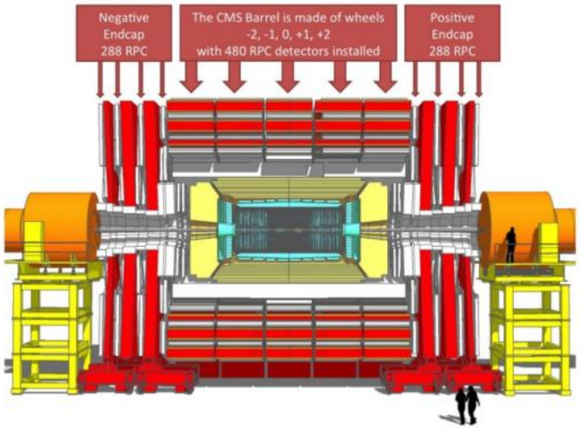


RPC Rates

TOWER
RB+2_far [Sector 4-9]
RB+2_near [Sector 1-3 & 10-12]
RB+1_far [Sector 4-9]
RB+1_near [Sector 1-3 & 10-12]
RB0_far [Sector 4-9]
RB0_near [Sector 1-3 & 10-12]
RB-1_far [Sector 4-9]
RB-1_near [Sector 4-9]
RB-2_far [Sector 4-9]
RB-2_near [Sector 4-9]
YEP3_far [Sector 4-9]
YEP3_near [Sector 1-3 & 10-12]
YEP1_far [Sector 4-9]
YEP1_near [Sector 1-3 & 10-12]
YEN1_far [Sector 4-9]
YEN1_near [Sector 1-3 & 10-12]
YEN3_far [Sector 4-9]
YEN3_near [Sector 1-3 & 10-12]

Barrel (RB rows) — **Endcap** (YE rows)

18 towers ->18 root files



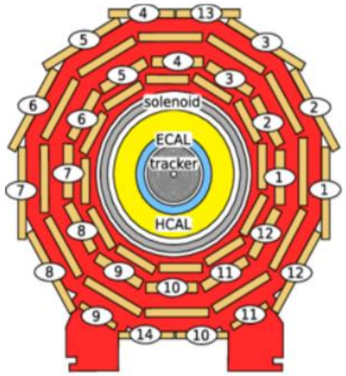
1056 RPCs in total

✓ Highlighted in red is the sign nomenclature used in CMS

✓ In violet the physical location of the linkboard

DPID	RUNNUMBER	EVENT	STARTTIME	STOPTIME	TDELTA	NCH	COUNT	COUNTNM
...	...	...	...	...	...	...	...	...

Barrel Wheel



Endcap Disk

entral-barrel-wheel-and-representative-endcap-disks_fig6_48926114

IBERO CIUDAD DE MÉXICO ® | Universidad de los Andes

RPC Rates: Lumi Fit

LHCFill	DPID	MIN_INSTLUMI	MAX_INSTLUMI	PO	PO_ERR	P1	P1_ERR	N	SIGNIFICANCE	STOPTIME
...	...	...	...	...	...	...	...	...	...	...

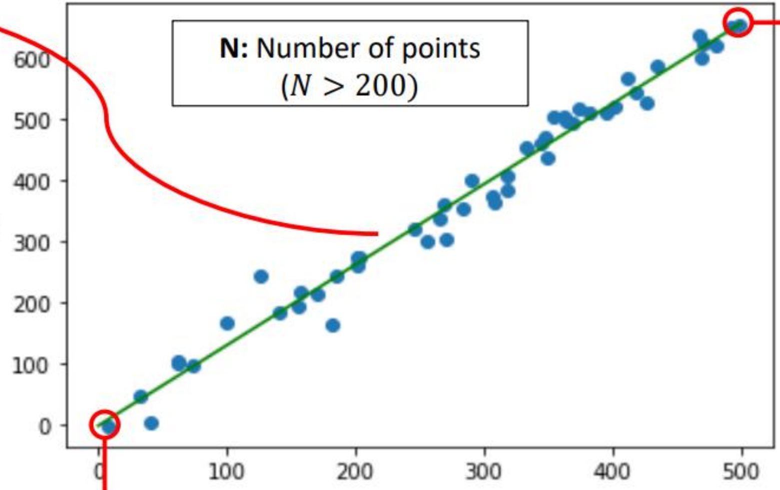
Linear fit:

PO = Intercept
$P1$ = Slope

PO_ERR = ΔPo

$P1\_ERR$ = ΔP1

N: Number of points
$(N > 200)$



Max_instlumi

Min_instlumi

Significance

Statistical test to determine whether the observed linear fit is statistically significant.

<u>More than 3000 fits are produced automatically per fill.</u>

Conclusions

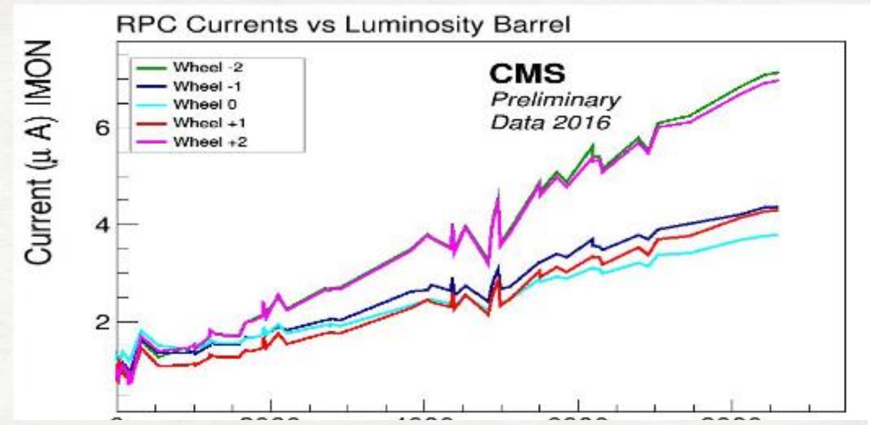
- **CMS RPC non-physics event data automation is extremely powerful JAVA-based framework which gradually evolved in the past 8 years.**
- Fully operational on production since 2018
- Continuous evolution on development ➔ GitLab project: https://gitlab.cern.ch/cms_rpc/rpc-automation
- **Ideology:** Data Synchronization, Data Tagging, Relational parameter studies, Full Circle in Automation
- **Methods:** Data Streaming Techniques, Multithreading, Block Averaging in Probe method, LHC Block conception, Luminosity methods
- Main condition parameters: detector currents and rates, environmental parameters
- Change of framework paradigm: move from hardware channels/sensors DPID to ChamberID.

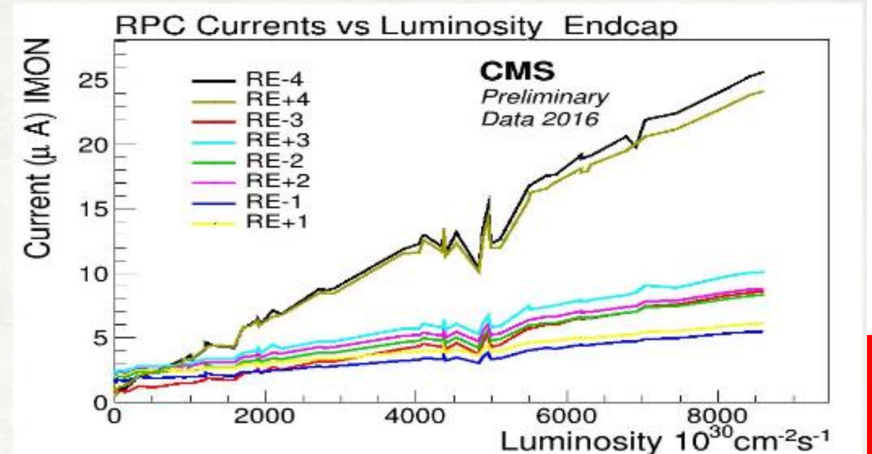
THANK YOU FOR YOUR ATTENTION!

Backup Slides

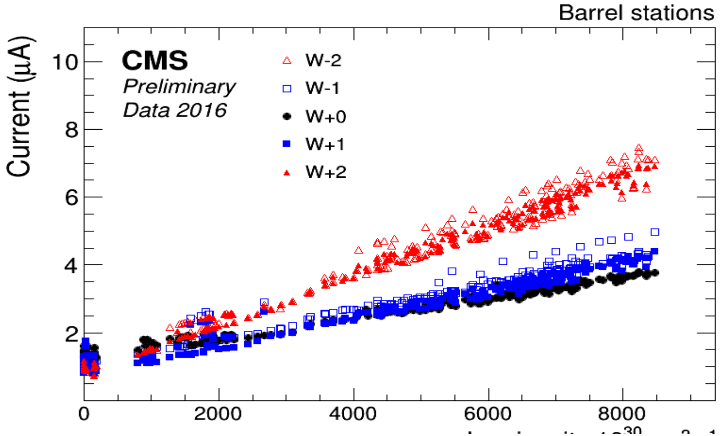
RPC Currents vs Instantaneous Luminosity Methods

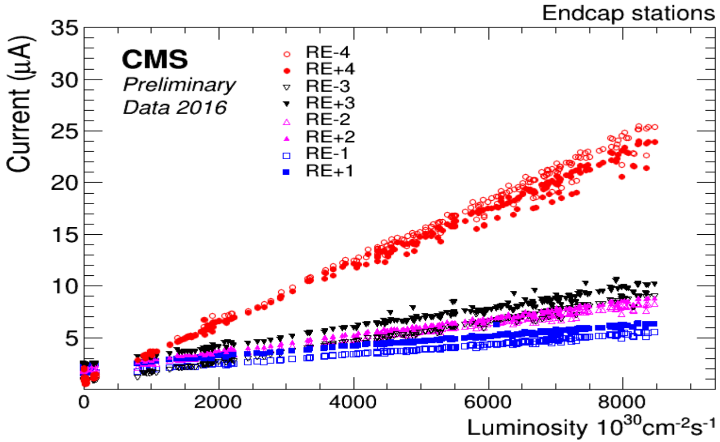
Current vs Luminosity Method 2





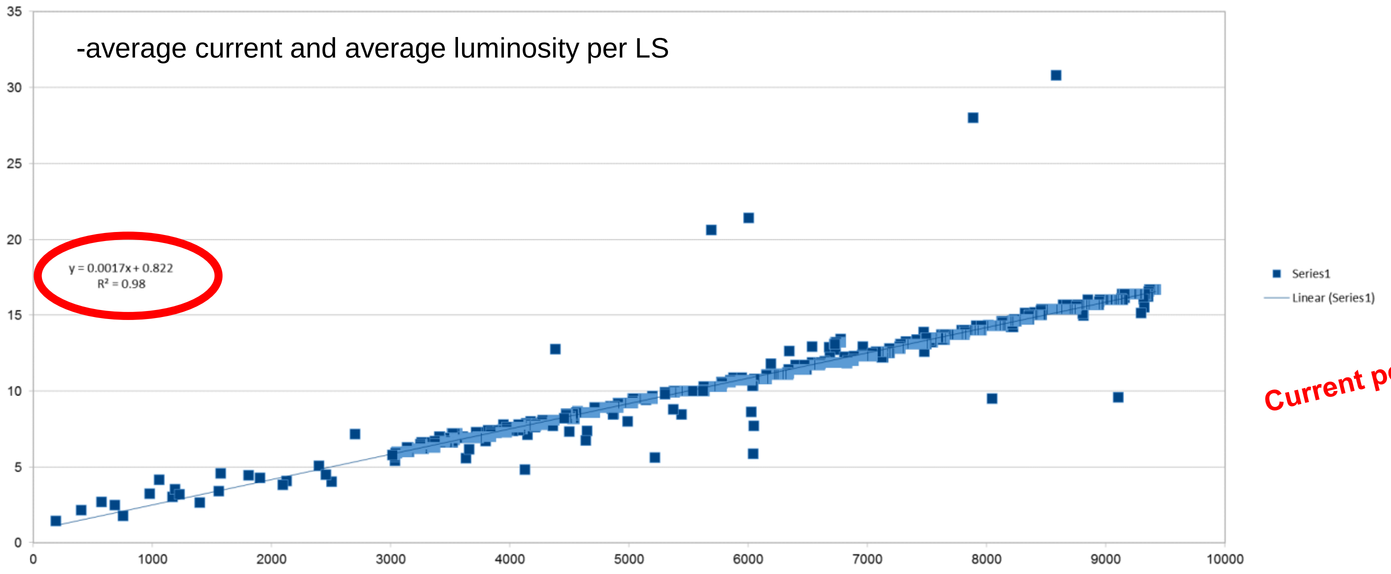
Current vs Luminosity Method 3





Are all these methods based on run averages the best?

NO → Going to Currents vs Luminosity Method4



Current per Lumisection doesn't exist !

Advantages:

- Fast (3-4 fills per year would be enough to have the same dependence)
- Do not average per run, uses every single LS
- Can depict very high instantaneous luminosities
- Could be used per DPID per fill, however …

So what? ➔ New streaming technique

$$\left(I_{mon}^{avg,\,LS_n}\right) = \sum_{j=1}^{l} \frac{I_{mon}^{j}}{T_{LS}} \Delta t_j \quad = \sum_{j=1}^{l} \frac{Q_{LS}^{j}}{T_{LS}} \quad = \frac{1}{T_{LS}} \sum_{j=1}^{l} Q_{LS}^{j}$$