Post-Mortem

J. Wenninger AB/OP

- Recap on PM
- PM event builder
- PM event analysis
- PM data storage

… with a look over the Accelerator / Research sector fence !

Motivation : the LEP2 example

The LEP2 performance depended largely on the RF system :
→ it was important to know about 'weak' RF components (cavities) !

When a RF unit tripped
- → beam loss.
- → sudden change of beam loading in all units.
- → more RF units trip !
  - It proved impossible to identify the unit that first tripped using conventional logging and monitoring (fast time scales).

For the last LEP run, the RF group build a small post-mortem system that could timestamp RF trips and beam loss with ~ few μsec resolution (GPS) :
- That system proved to be a crucial diagnostics and greatly enhanced the LEP efficiency for the last 'Higgs-hunt' run !

LHC stored energy : new territory

Stored magnetic energy : 10.8 GJ ⟶ Energy extraction

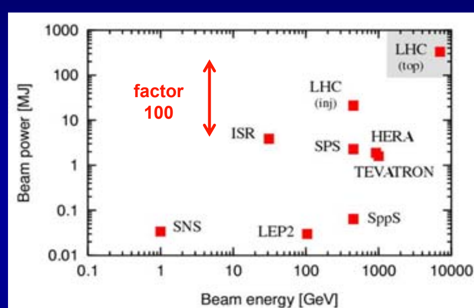
Stored beam energy :
0.35 GJ / beam

sufficient to melt
500 kg of Cu

equivalent to
- 90 kg of TNT
- 25 kg of sugar

↓

The beam dump is the only component able to absorb the full 7 TeV beam.



The LHC must be protected against damage due to uncontrolled energy release of any form !

The post-mortem system

- The LHC will be protected by over 10000 interlock channels :
  - thousands of quench detectors
  - 3000 beam loss monitors
  - …
- The LHC has a minimum cycle of ~ 2 hours (7 TeV back to 7 TeV)
  → learning by trial and error is very inefficient !

For that reason we need a DIAGNOSTICS tool :

- To understand when, why and how interlocks are triggered.
- To determine the initial cause of a 'problem', to adjust interlock thresholds… we must be able to see the last moments before the beam disappears in the dump block !

This tool is the Post-mortem system !

Post-mortem ingredients

- Every LHC equipment and diagnostics system must implement a circular PM buffer of appropriate depth holding the latest data (example : last 1000 turns for beam instruments …).
- Data must be time-stamped to ~ ms or µs depending on type.
- The PM buffer must be frozen by an external post-mortem event or by self-triggering.
- The PM data must be combined to form the post-mortem event data : size ~ few Gbytes.
- The PM data must be automatically analyzed. ‘Digested’ information must be generated for operations.
- The PM data must be stored – the most relevant data must be stored for the lifetime of the LHC. Some of it will be important for INB.

Post-mortem data

- Data sent to the PM system should be self-describing.
- Sharing the same data format with the logging system is probably a good idea.
- The LHC experiments use a data format provided by the root C++ package (compressed & system independent encoding).
- An ‘event builder’ is required to :
  - assemble the data (push or pull ?).
  - assign it a unique PM event number (key).
  - verify data integrity and completeness.
  - store the data on disk for immediate analysis.
  - possibly send it to long term storage.

PM operation modes

The post-mortem system has 2 basic operation ‘modes’.

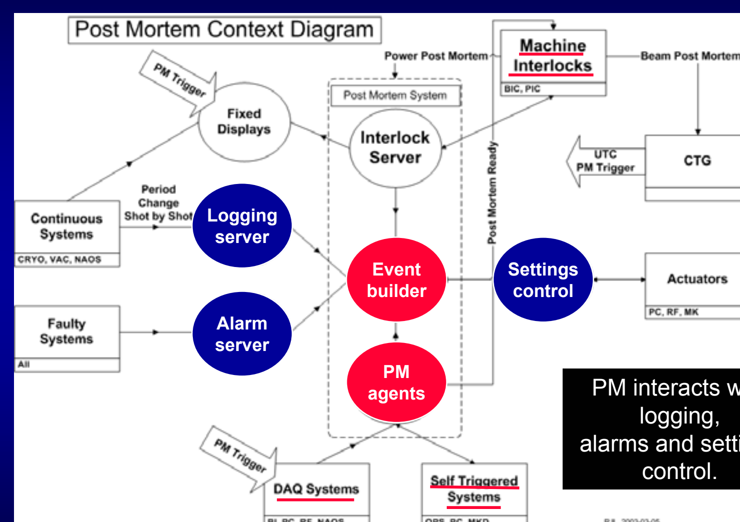
Operation without beam :

- Each powering sub-sector must be handled as an entity that can have an abort independently of all other sectors.
- Main systems : power converters, magnet protection, interlocks (too a lesser extend vacuum & cryogenics).
- In this mode the systems are self-triggering.

Operation with beam :

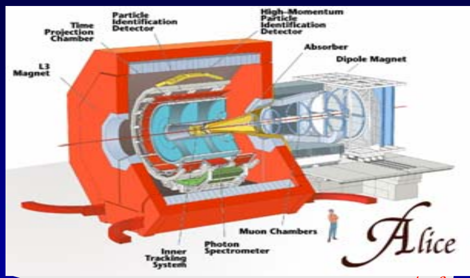
- The machine has to be considered as a whole.
- All equipment systems are involved.
- PM is triggered over the interlock system (most likely timing event).

PM, Logging, Alarms

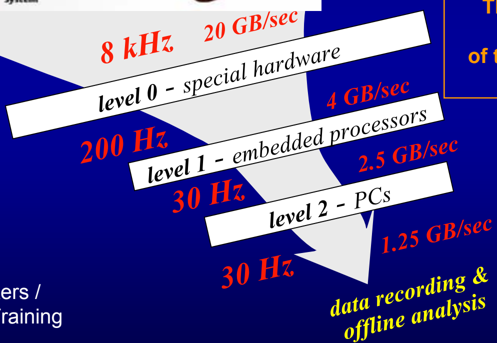


PM interacts with logging, alarms and settings control.

ALICE DAQ



Independent detectors	**12**
Total weight	**10,000t**
Overall diameter	**16.00m**
Overall length	**25m**
Magnetic Field	**0.4Tesla**



The experiment trigger is the 'equivalent' of the machine protection system

F. Rademakers /
CERN Acad. Training

Mass storage @ LHC

			Readout (Events/s)	Readout (MB/s)	Event size (MB)	Data archived /year (PBytes)
	ALICE	Pb-Pb	10^2	1250	12	2.5
	CMS	pp	10^2	100	1	6.0
	ATLAS	pp	10^2	100	1	3.0
	LHCb	pp	$2x10^2$	40	0.2	1.0
	PM		1000/year		~5000 (raw)	0.005

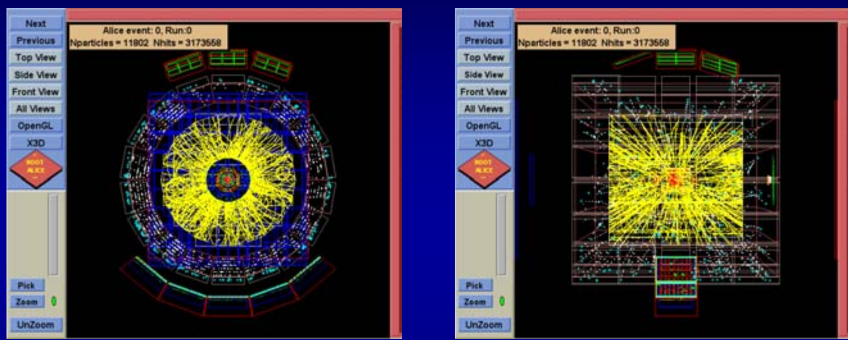
Data Storage

- I presume that regular LHC logging goes to ORACLE…
- All LHC experiments write their RAW data to tapes :
  - Tape handling is done via CASTOR, a disk pool manager coupled to tape storage. Handling of tapes is transparent to the user who sees a UNIX-like file system. Data is accessed over disk cache.
  - Typical tapes hold ~ 100 GB – not really competitive compared to disks, but simpler to handle (robots) !
- Both solutions could be OK for long term storage of PM data, but :
  We need intelligent storage of complex data like the last 1000 beam turns for various instruments in we use ORACLE.
- To improve performance for analysis of PM events just after an abort
  → run from a local disk !

PM event storage

- The RAW data volumes for each PM event are very large :
  - some GBytes !
  - depends strongly on amount of bunch-bunch and turn-by-turn info (RF is the dominant client !).
- The experiments have much smaller event sizes – but they only record 'useful' info :
  - channels without signals (hits) are not included.
- We must at some stage decide if we archive ALL raw data.
- We could envisage to store in easily accessible form (ORACLE DB or disk) only useful / compressed information.
  For example : for a PC that had no problem, keep only state + min/average/max current (set & read), instead of the full data !
  ……

A Pb-Pb collision in ALICE



To extract any useful info, such a simple event display is not sufficient !

A reconstruction / analysis code is required to group detector signals into tracks, energy clusters and eventually reconstructed particles !

PM analysis

The situation for PM is similar to the experiments data :

- We want to find the relevant info in the big byte-mess !
- We must scan for faulty channels and states, summarize the beam evolution just before the abort (orbit change, beam loss evolution and location, loss rate change…)….
- The summary info must be 'presented' to operations for guidance. The operators cannot be asked to browse the data with JAVA guils !

A considerable and also complex effort : therefore …

- A modular design is needed, that can incorporate modules prepared by various people in a variety of languages (C, C++, JAVA….).
- The code must be able to evolve rapidly as experience is gained.
- It must be possible to re-run the analysis on past events.
- The output information must also be stored !!

PM analysis (cont'd)

Also :

- The data volumes are large and time counts. 'Response' within minutes please !!
- The code must be fast and so must be the I/O !

Finally :

- For the beam dumping system, a PM analysis is mandatory before any beam can be re-injected again into the ring.

Summary

- To operate the LHC efficiently we need a PM system.
- I predict that PM will be among the hottest stuff in the LHC !
- At the equipment level, work is 'in progress' – but we must get going soon with the higher level PM (data collection & analysis) :
  - We need some help – volunteers are preferred !
  - A couple % of J. Wenninger and R. Lauckner is not sufficient in the longer term…
- Milestone # 1 : sector commissioning in 2005 !

More details on PM can be found in LHC Note 303.