# The High-Level Trigger of the ALICE Experiment at CERN - An overview

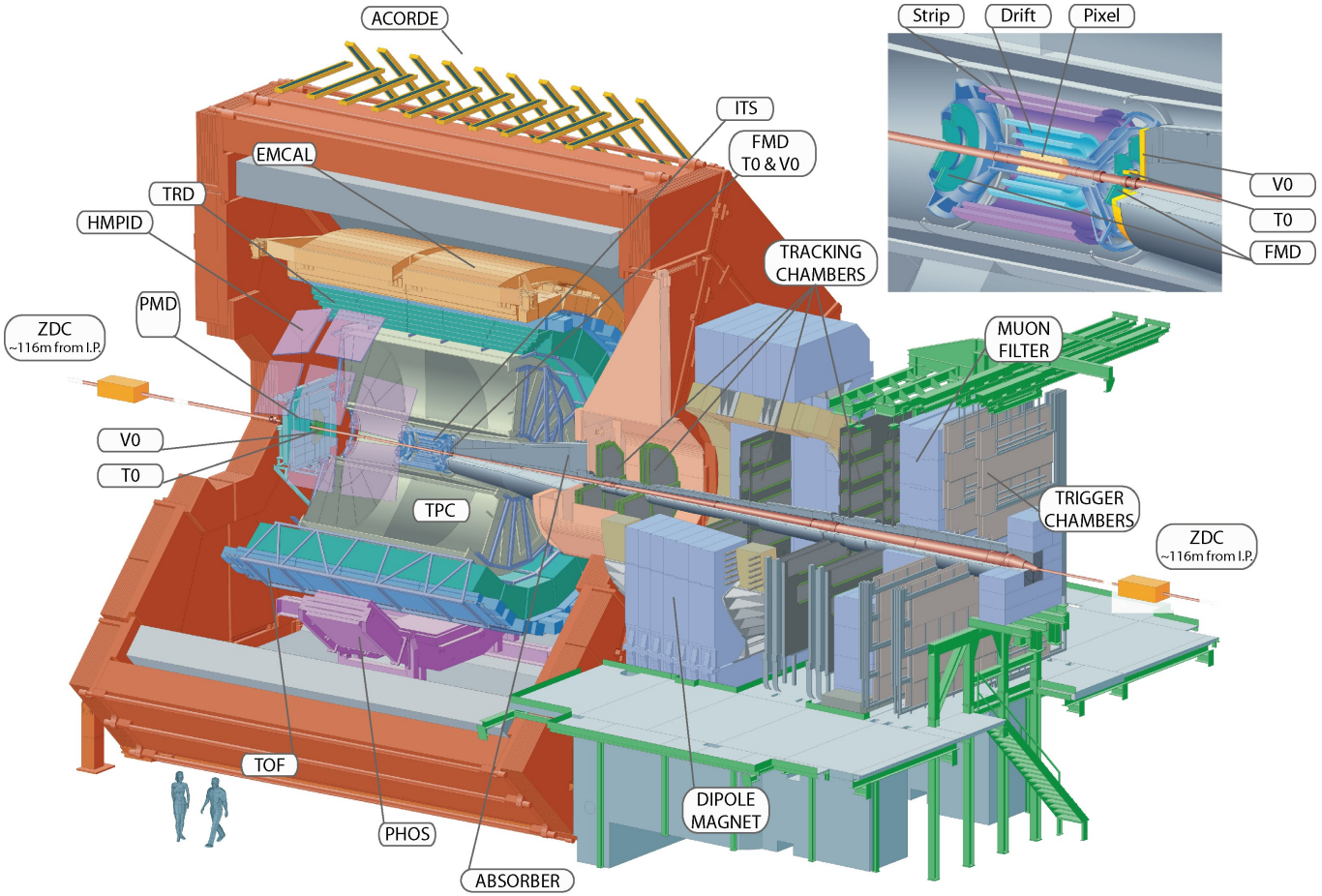
Dinesh Ram
Dinesh.ram@cern.ch
University of Bergen, Norway

# The Why? What? And How ? Of ALICE-HLT

# Introduction

- ALICE is the acronym for A Large Ion Collider Experiment.
- It is one of the 4 larger experiments of the Large Hadron Collider (LHC)
- It is a dedicated detector system designed to study the interaction of heavy-ions at LHC energies.
- The formation of a new phase of matter - the quark-gluon plasma, is expected.
- The ALICE detector system consists of 18 different sub-detectors.
- ITS, TPC, TRD, EMCAL, PHOS, PMD, MCH & MTR, ZDC etc.

# THE ALICE EXPERIMET @ CERN



# Some numbers

- Heavy-ion interactions are characterized by very large multiplicities of about 15000 particles per event.
- Approximately 70 MB per event @ ~ 200 Hz. Corresponds to about 25 GB of data per second to be handled.
- The Time Projection Chamber (TPC) detector is the largest contributer to this volume of data.
- Maximum available data-storage bandwidth in ALICE is 4 GB/sec.
- So, how to handle this colossal amount of data?

# The High-Level Trigger

- The High-Level Trigger (HLT) is the last trigger stage of ALICE.
- Does a complete, fast online reconstruction of detector data to decide if an event is “interesting” and if it should be stored for further offline analysis.
- Also performs compression of event data without loss of physics information.
- Performs online monitoring of detector data (Refer to Hege Erdal's Talk)
- Event reconstruction places a high computational load on the HLT.
- It is a large, dedicated high performance computing cluster.

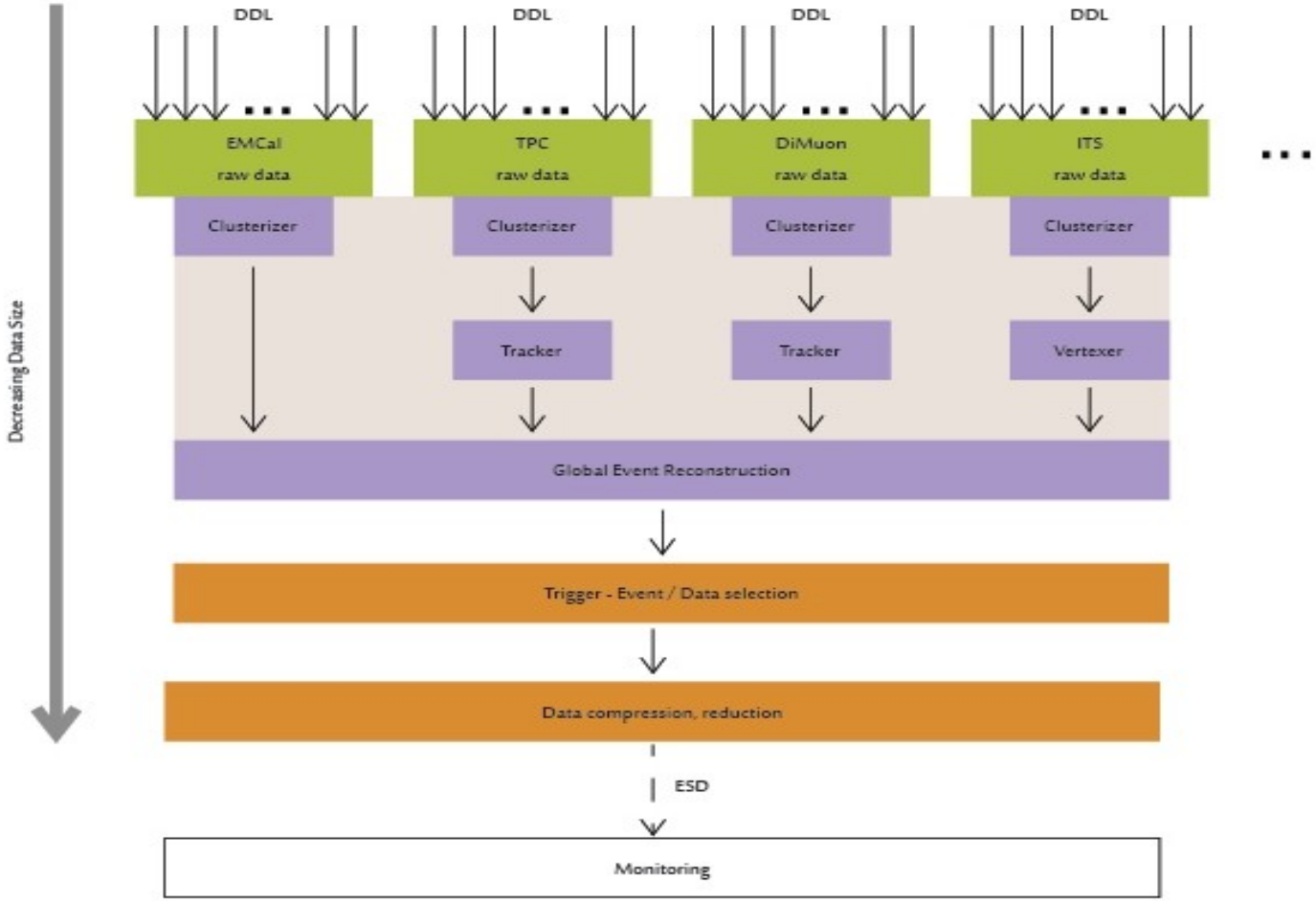
# HLT Cluster at Point 2 - CERN

- ~ 250 nodes
- 2752 CPU cores
- 64 GPUs
- 246 FPGAs
- 5.29 TB physical memory
- Fast infiniband network
- Linux operating system
- Fraunhofer File System (FhGFS)

# Architecture

- The reconstruction follows a hierarchical pattern which emulates the physical layout of the detectors.
- First-level reconstruction starts at Front-end processor (FEP) nodes which host the PCI-X, HLT Readout Receiver Cards (HRORCs).
- The HRORCs receive data from the Data Acquisition System (DAQ) system.
- FPGAs hosted on the HRORCs perform pre-processing of the data before copying it over to the main memory of the FEP machines.
- Subsequent reconstruction levels - the event-by-event analysis, event-selection and data-compression - are processed on the Computing Nodes following a well-defined data flow hierarchy.

# Data-flow in the HLT




Readout data passes through the cluster in several steps of analysis and merging.

# HLT Data-transport framework

- Goal is to be able to use maximum number of CPU cycles for analysis, while minimizing the CPU usage during transfer of data between components running of different cluster nodes.
- The data transfer inside the cluster has to have a minimal latency to handle the high data rate.
- A software, data-transport framework was developed to handle the transfer of event data through the HLT cluster.
- Highly efficient and flexible.
- Modular architecture, where a number of independent software components which can communicate via a common interface.
- Components are plugged in together in different configurations satisfying different physics requirements and run conditions.

# HLT Software Chains

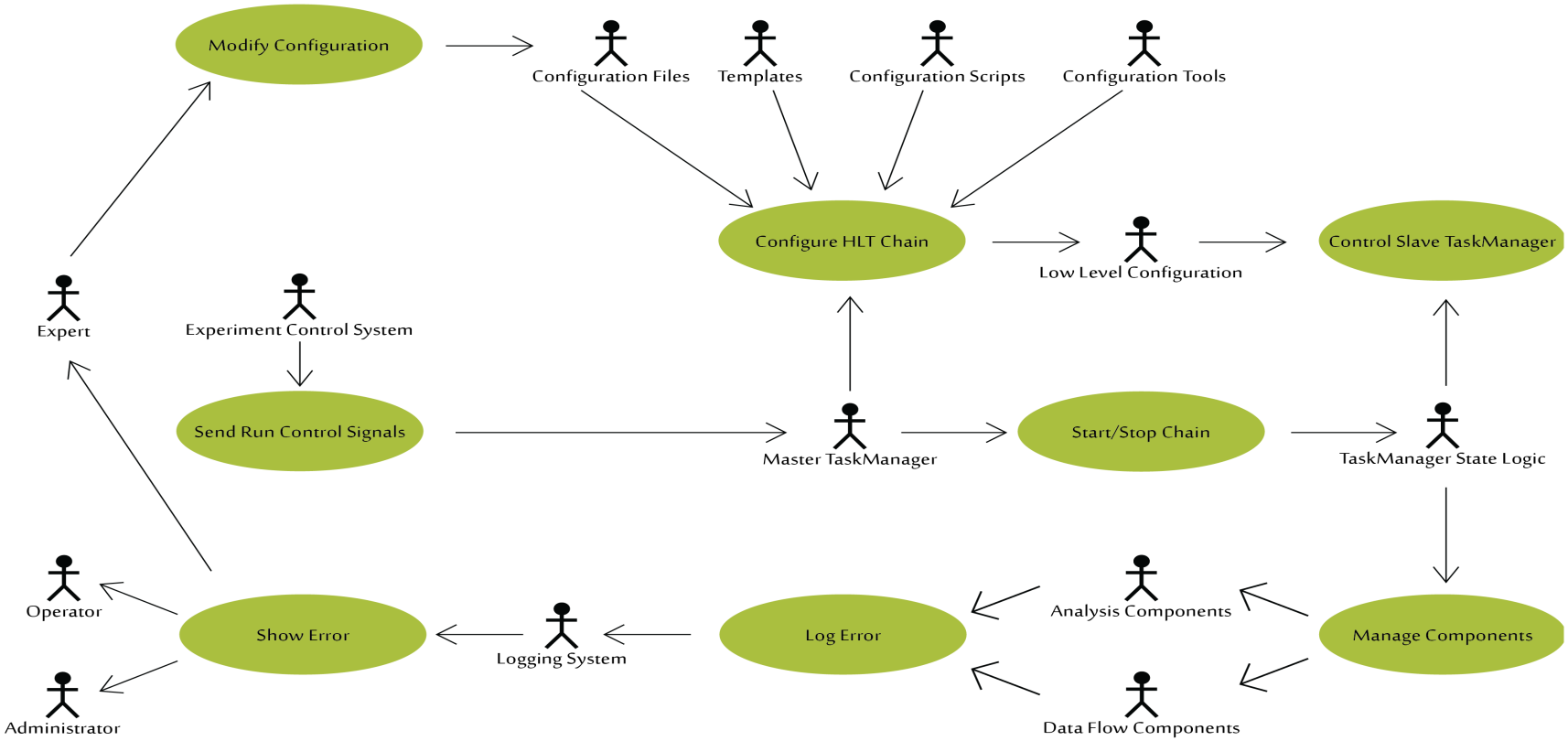
- Different analysis components are “chained” together to form a logical data-flow pattern in the HLT.
- An XML based configuration scheme known as the Simple Chain Configuration (SCC) is used.
- Create a configuration describing the application specific components only (data sources, processor components, data sinks). Data flow components will be created automatically.
- This scheme allows to specify details like the name of the component, the library to be used, the multiplicities and the machine in which the component is supposed to be running.
- Example :

```
<Proc ID="GDL0" type="prc">
   <Cmd>DummyLoad</Cmd>
   <Parent>DL0</Parent>
   <Shm blocksize="12k" blockcount="1024" type="sysv"/>
   <Multiplicity>1</Multiplicity>
</Proc>
```

# The TaskManager Control system

- The state-logic for the entire system is provided by the TaskManager control system which implements a state-machine.
- It controls a few thousand processes distributed over all nodes in the cluster using a control hierarchy.
- TaskManager coordinates when the state-change commands are sent to Controlled components
- It ensures that the entire system is in a consistent state by querying the components at regular intervals.
- Distributed management paradigm with one “slave” TaskManager process per node, controlled hierarchically by higher level TaskManagers.

# A picture of the system during online chain running



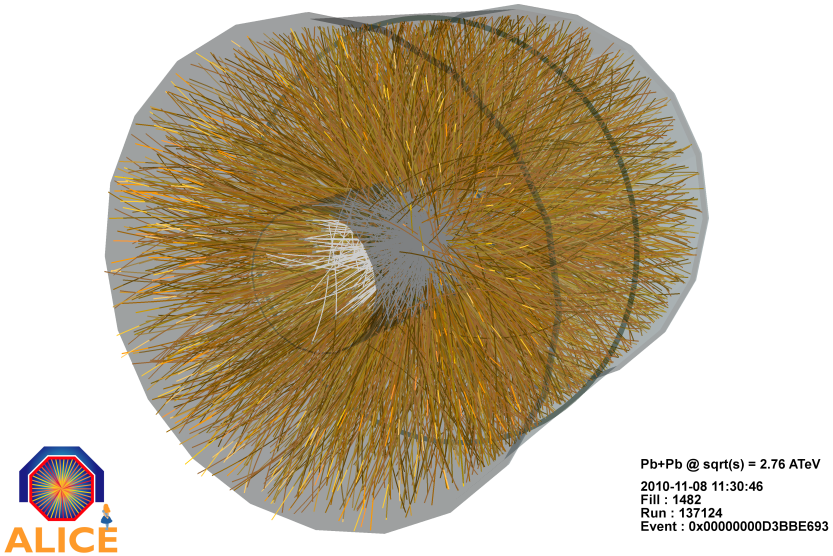
# Cluster Management

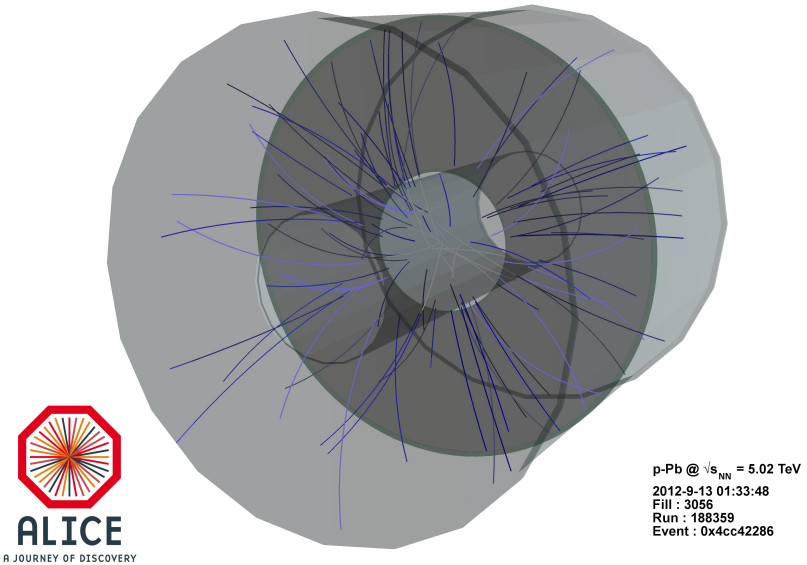
- Efficient monitoring and management of all the nodes in the cluster is of prime importance to a large system like the ALICE HLT.
- Cluster monitoring tools – Ganglia, SysMES
- SysMES clients installed in all machines actively monitor the system parameters – system temperature,  network connectivity, memory and usage, switch config changes, mount failures etc.
- Can take certain actions independently based on well defined SysMES rulesets.
- CHEF is used for install and maintain different node categories by assigning roles to them.

# Running the HLT in the 2011 Heavy-ion period of ALICE

- The average data rate processed by the HLT was 7GB/sec in 2011 heavy-ion period.
- Each instance of the processing chain approximately ran 40-50 different physics analysis components with varying multiplicities for load balancing purposes.
- A total of approximately 1600-1800 processing instances and 9000-10000 data-flow instances, distributed over the cluster nodes.
- Each analysis process instance is roughly assigned one CPU core and 2 GB of physical memory.
- The TPC slice tracking algorithm was run on GPUs.
- An optimal load-balancing strategy is non-trivial.

# Online Monitoring in the HLT – Pretty pictures



Questions ?