Argonne NATIONAL LABORATORY

... for a brighter future

DHCAL Software in CALICE DAQ Framework

J.L.Schlereth

December 7, 2007

U.S. Department of Energy

UChicago Argonne

A U.S. Department of Energy laboratory managed by UChicago Argonne LLC

Overview

- Introduction
- Basic CALICE DAQ concepts
- CALICE DAQ Infrastructure
- DHCAL extensions
- Summary

Introduction

- What is CALICE DAQ?
  - framework for DAQ applications written in C++
  - intended to be lightweight and transportable (among sites)
  - implemented for Linux only
  - distributed as source code, only external libraries needed are ROOT and HAL
  - designed around Calice Readout Card (CRC)
  - ... but can be extended to support other hardware
- More information

http://www.hep.ph.ic.ac.uk/calice/daqSoftware/daqwup/daqwup.ps

Basic CALICE DAQ Concepts

- Records
  - generalized events or occurrences within DAQ system
  - contents completely user defined
  - consists of short header followed by arbitrary amount of data
- Sub Records
  - can be inserted into records
  - sub record type associates payload with data storage class
- Data Storage Class
  - define actual structure of sub record payload
  - access and insert methods exist to manipulate data storage objects
- Modules
  - execute user defied actions on records
  - examples: configuration, event readout, monitoring, file I/O
  - aggregate modules perform actions on groups of modules
- HAL::VMEDevices
  - based on concept of bus adapters and address tables which aid in portability
  - user defines classes derived from HAL base class that have many methods to operate DAQ

CALICE DAQ Infrastructure

- Run Control
  - record types represent transitions in the DAQ state machine
  - record is created for each state transition and passed to modules which take action based on type
  - inter process communication has been implemented to allow DAQ to run in background
- Monitoring
  - separate applications can sample records from DAQ application
    - *shared between processes on same node using shared memory*
    - *shared between processes on different nodes using sockets*
    - *histograms can be accessed by ROOT using memory mapped file*
- Data Recording
  - support for reading/writing ascii or binary files
  - implemented as a module, just as a readout or monitoring module

DHCAL Extensions

- Data Storage Classes
  - DhcBeConfigurationData – contains data for the back end section of the DCOL
  - DhcFeConfigurationData – contains data for the front end, i.e DCAL ASICs
  - DhcReadoutConfigurationData – contains parameters controlling the readout
  - DhcEventData – contains the timestamps and hit data
  - TtmConfigurationData – contains settings for the TTM board
- Modules
  - DhcReadout – performs slow control functions and event readout through DCOL
  - TtmReadout – performs control functions for the trigger through TTM
  - DhcConfiguration – loads the XXXConfigurationData objects from external DB
- HAL::VMEDevices
  - DhcVmeDevice – performs operations on the DCOL and FE ASICs through DCOL
  - TtrmVmeDevice – performs operations on the TTM

Summary

- Development Cost
  - estimate ~6 FTE weeks to develop and test extensions to get working DAQ
  - estimate ~1-2 FTE weeks to develop applications which convert to data formats currently used in offline analysis
- Configuration Issues
  - can current SQLite database be extended,integrated into this framework?
- Benefits
  - extensions provide reusable code which can be deployed in many applications
  - CALICE DAQ infrastructure provides useful features for both online and offline analysis
- More information

  http://www.hep.anl.gov/jls/dhcal/CALICE_DAQ_Design.pdf