Control Configuration and Monitoring

Pierre Lasorak





- CCM is responsible for controlling, configuring and monitoring the DAQ
- DAQ is a complex system:
 - Many different types of hardware need to be handled in a coherent manner (FELIX, Timing, DAPHNE...)
 - Ever increasing complexity of applications, configuration, performance optimisation \bullet requirement, and diverse users platforms
- We already need to be able to run with high up time for cold boxes and soon ProtoDUNE2, whilst developing new control, deployment and configuration systems.









- Responsibility of the Run Control
- Rewritten since ProtoDUNE1, along with the whole DAQ framework
- Use an in-house control software called nanorc (Not ANOther Run Control):
 - "Lightweight" control software written in python
 - Controls standardised DAQ applications (C++)
- Over the past year:
 - Uses external services (run number, registry, logbook, configuration)
 - Final state machine
 - Production and test environments
 - Global timing partition

Control









Run control / Process management

- Traditionally use SSH to spawn DAQ processes on any host of the network
- Work over the past month to prototype the use of Kubernetes (k8s) in the RC:
 - Google open source resource management software, typically used in data centres
 - Packages up DAQ applications into Docker images
 - K8s are able to understand resources, connected hardware, load on a node, ...
 - Distributes workload on nodes
- Kubernetes is expected to operate ProtoDUNE2 with it by next release (v3.1)



NanoRC WebGUI												
Control tree	Controls:											
conf-flx-emu-wib2-	State control buttons:				Present state	:						
	boot terminate					none						
	Last response	from nanorc:										
	Selected: conf-flx-emu-wib2-opmon-k8s											
	Name			State	Host	Last cmd	Last succ. cmd					
	conf-flx-emu-wi	none										
	conf-flx-em	u-wib2-opmon-k8s	none									
(A) Inches and the s		eesk –	0 6000	-h-								
kubernetes	plas	orak 👻	Searc	n								
\equiv Workloads > P	ods											
	_											
Workloads 🔳	Po	ds										
Cron Jobs		Name			Labels							
Daemon Sets Deployments		trigger			app: trigger							
Jobs		dfo			ann: dfa							
Pods					арр. 010							
Replica Sets		dqm0-df			app: dqm0-df							
Replication Controllers		dqm0-ru			app: dqm0-ru							
Stateful Sets		hsi			app: hsi							
Service N		nufly()			app: rufly0							
Ingresses Services					app. runxo							
Config and Storage		dataflow0			app: dataflow0							
g anna a tar aga												



Run Control / Kubernetes

- Continuously implementing improvement towards better integration of the k8s in the DAQ:
 - FELIX hardware discovery service
 - Controllers
 - Storage for raw data
 - Microservices
 - Timing system
 - Monitoring
 - \bullet . . .







- JSON messages used to configure the DAQ applications
- Efforts to improve the configuration generation: \bullet
 - Better representation of the system in the configuration generation tools - Phil Rodrigues
 - Impact of the K8s
 - Current "push" method:
 - Stored in files (can also be stored in a DB)
 - Run control reads in the configuration, and sends data to the \bullet application with all the configuration data
 - Future "pull" method:
 - Run control makes the configuration "available" to the apps, and lacksquarethe apps are responsible for getting it
 - Towards virtualisation of the configuration, with a Data Access Layer

Configuration







- Quite a lot of work to integrate the monitoring of different part of the system
 - DAQ application monitoring \bullet
 - Trigger, FELIX, Readout, timing...
- DAQ applications send metrics to an Influx DB
 - Metrics can be displayed on a Grafana dashboard
- DAQ applications send error to a Kafka broker, which forwards them to a Postgres DB
 - Again can be seen in Grafana dashboard

Monitoring





• Main, overview Grafana dashboard:

• Many more subsystem dashboards...

Pierre Lasorak

Monitoring

					2 Last 5 minut	tes ~ Q	ις 22 - Δ
					\equiv Subsyste	ms \equiv Global	品 NP04 Network
ne	Run number			Issued triggers	Written TRs		
min		5		290		28	3
				Message Reporting			
	se' 🖓	time 🖓	applica 🖓	message 🖓		issue_name 🖓	' host_name
	INFO	2022-06-23 15:52:07	trigger	Trigger is active now		trigger::Trigger	trigger
	ERR	2022-06-23 15:52:06	ruflx0	Unexpected chunk size: 5668	flxlibs::Unexpe	rufix0	
	ERR	2022-06-23 15:52:06	ruflx0	Unexpected chunk size: 5668			ruflx0
	INFO	2022-06-23 15:52:06	trigger	Start of run 5		trigger::Trigger	trigger
5:56:00							

Conclusion

- CERN-NP04 environment.
 - Virtualisation / containerisation of the "ingredients" of CCM
- A lot of work over the past months to support the cold box runs and develop new features:
 - Run Control
 - General development, Web GUI ullet
 - RC services
 - Kubernetes \bullet
 - Configuration ullet
 - Tools ullet
 - Pull model / Data access layer \bullet
 - Monitoring
 - Centralised dashboards

• Slowly entering a phase of consolidation of the CCM systems, and making them available outside of the