ToolDAQ DAQ Framework

Dr Benjamin Richards (b.richards@qmul.ac.uk)

Overview

- ToolDAQ description
- Hardware Testing
- The Annie ToolDAQ
- Hyper K GPU

ToolDAQ

ToolDAQ is an open source DAQ Framework developed here in the UK.

It was deigned to incorporate all the best features of other frameworks whilst:

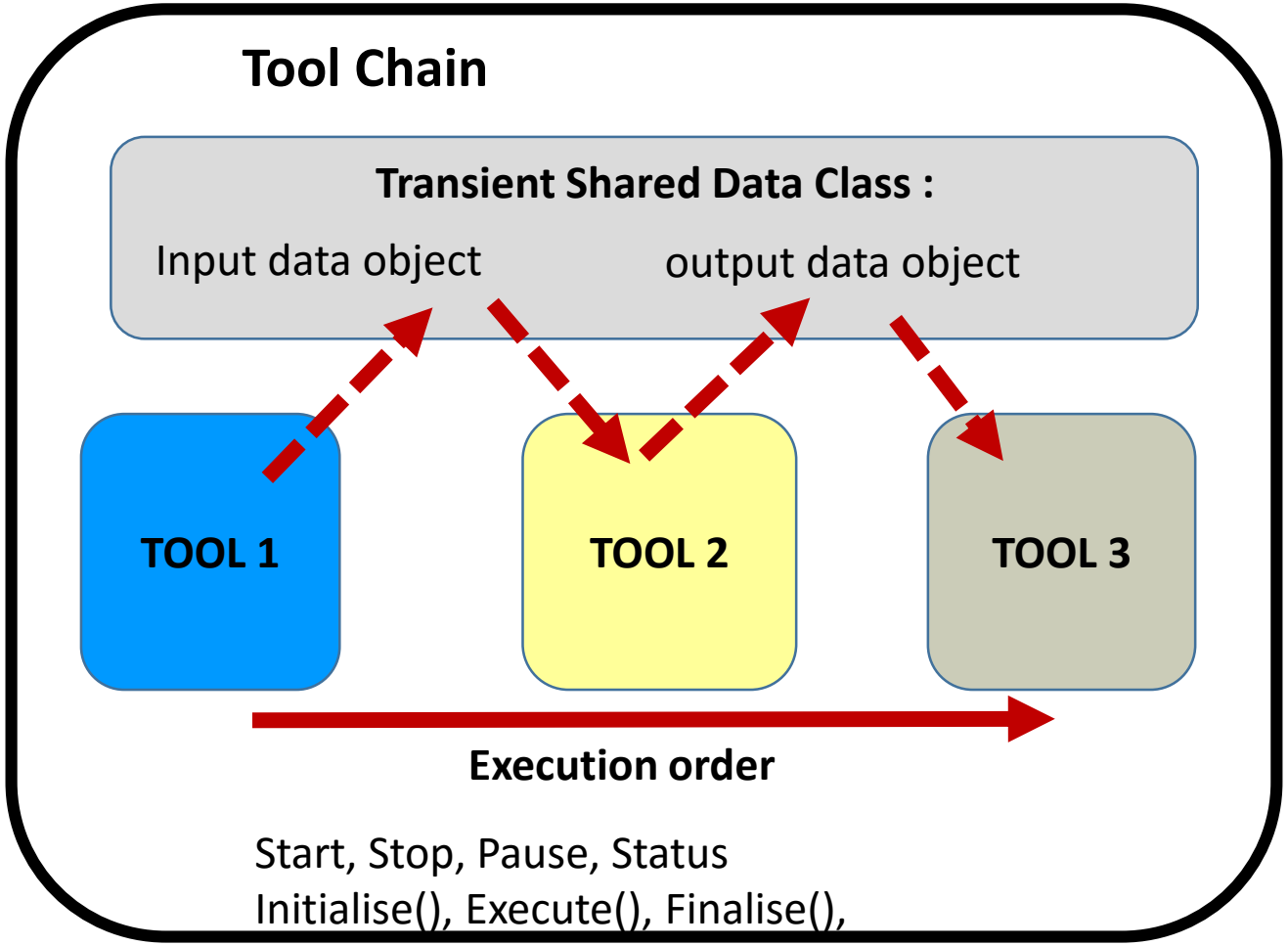
1. Being very easy and fast to develop DAQ implementations in a very modular way.
2. Including dynamic service discovery and scalable network infrastructure to allow its use on large scale experiments.

Features

- Pure C++
- Fast Development
- Very Lightweight
- Modular
- Highly Customisable / Hot swappable modules
- Scalable (built in service discovery and control)
- Fault tolerant (dynamic connectivity, discovery, message caching)
- Underlying transport mechanisms ZMQ (Multilanguage Bindings)
- JSON formatted message passing
- Few external dependencies (Boost, ZMQ)

ToolDAQ a simplified Look

- At its simplest **ToolDAQ** Allows users to create modular classes Known as **Tools.**
- These **Tools** can be added together into sequential collections of **Tools** known as **ToolChains** .
- These **ToolChains** handle all the more complicated aspects of a DAQ system , like DAQ remote control, service discovery, status whilst allowing for easy sequential **Initialisation, Execution** and **Finalisation** of the contained **Tools.**
- The **ToolChain** also contains a transient **DataModel** class for storing and passing data between tools

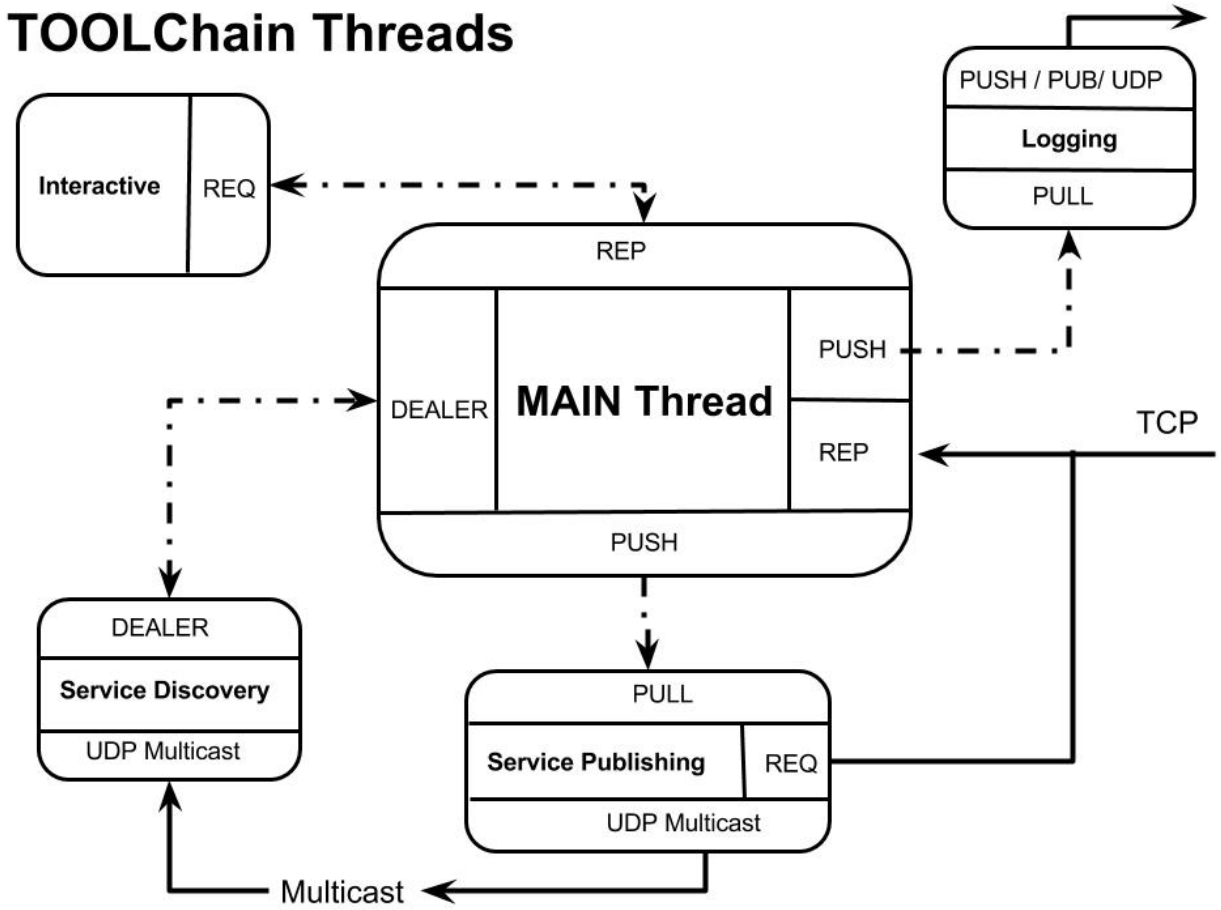


ToolDAQ in Detail

All Users need do is write their own modular Tools for hardware control and data processing while under the hood the software provides a powerful set of features out of the box.

- Three execution modes (Interactive, Inline and Remote)
- Built in distributed Network DAQ control through command line or Web Interface
- Built in Dynamic Service Discovery and Publishing
- Remote or Local Logging modes
- Simple threaded scalable Fault tolerant NtoN Networking technology provided by ZMQ
- Configuration file tools using a universal data storage class

And more…



Implementations

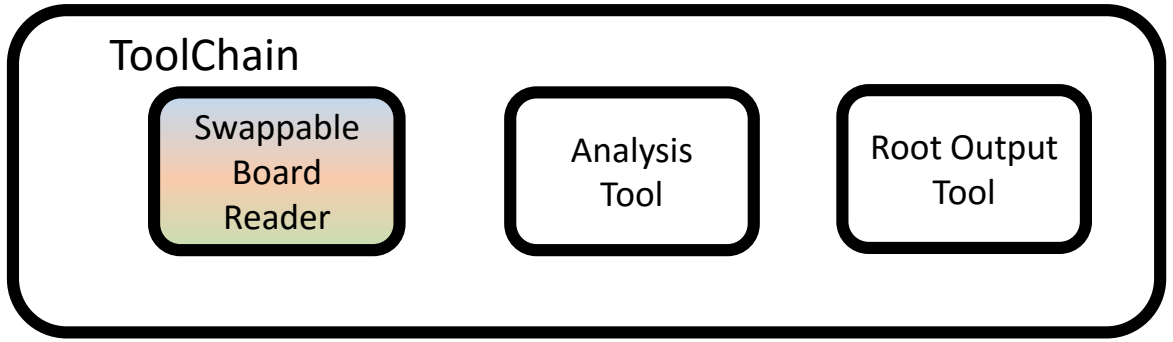
Hardware Testing

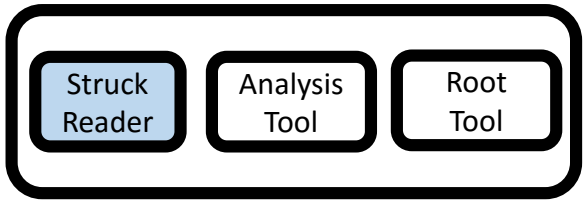
ToolDAQ was originally born out of a need to test multiple Hardware sources quickly and efficiently and being frustrated with the complexity, slowness and lack of customisability to develop these with existing frameworks.

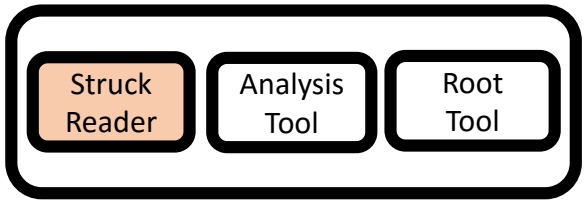
- At QMUL we have a hardware test stand and many different hardware components that needed testing
    - PSEC LAPPD front end boards
    - Custom FPGAs
    - ADC and TDC cards
    - VME / Camac
    - VME CPUs
    - Rack mounted Servers

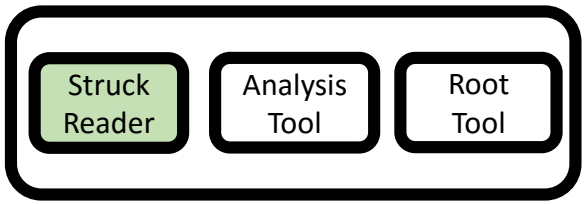
Modular simplified development

Using the **modularity** of ToolDAQ's Tools I was able to write a single application and hot swap Tools to analyse and recode the output form multiple data sources quickly.



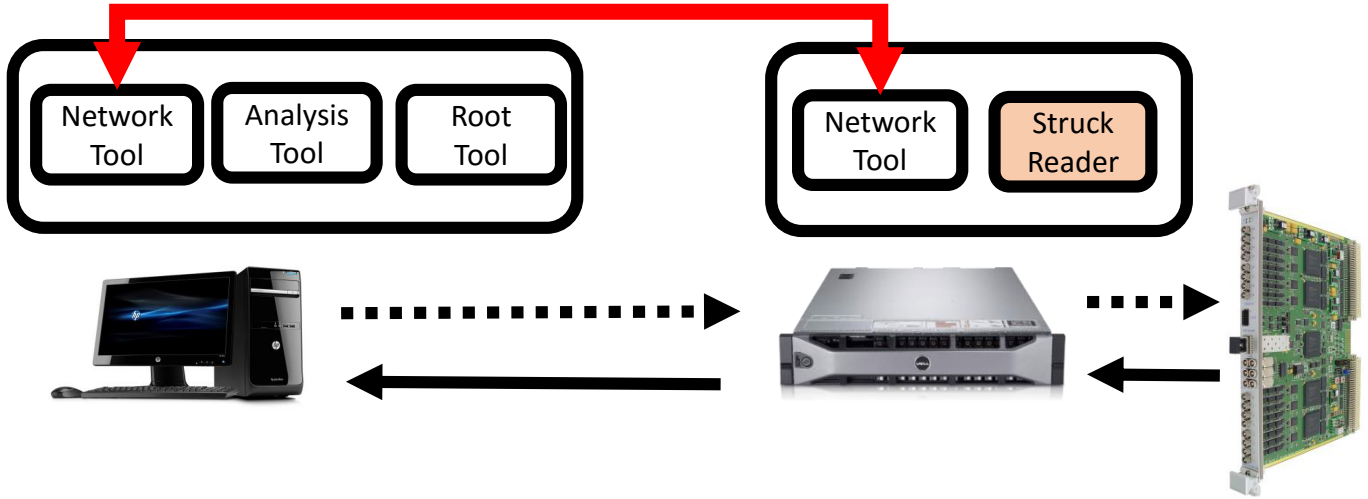






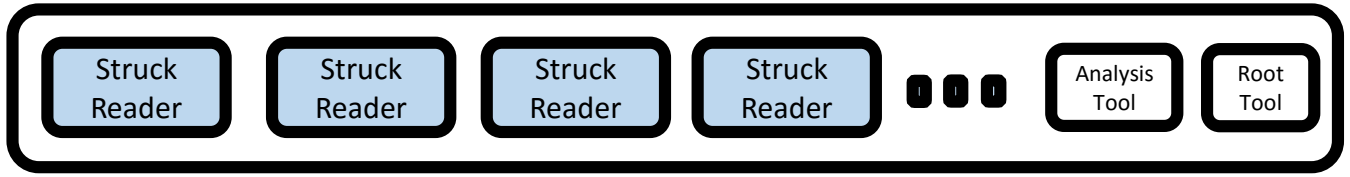
Expanding Functionality

From simple beginnings the built in functionality allowed expanding usage.



1) Remote Network Operation

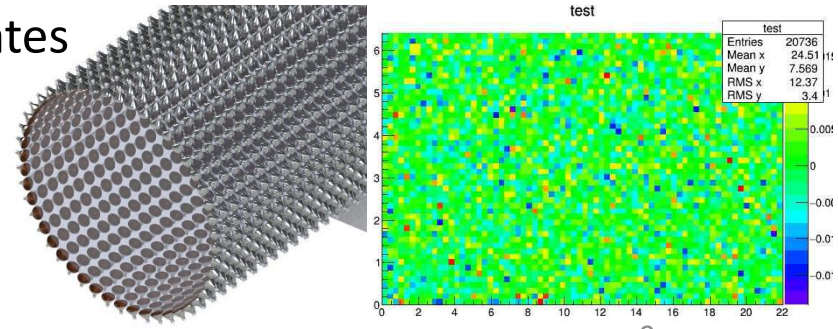
The built in Remote control tools allowed with no modification these implementation to work over the network from a remote location



2) Simulate multiple hardware sources

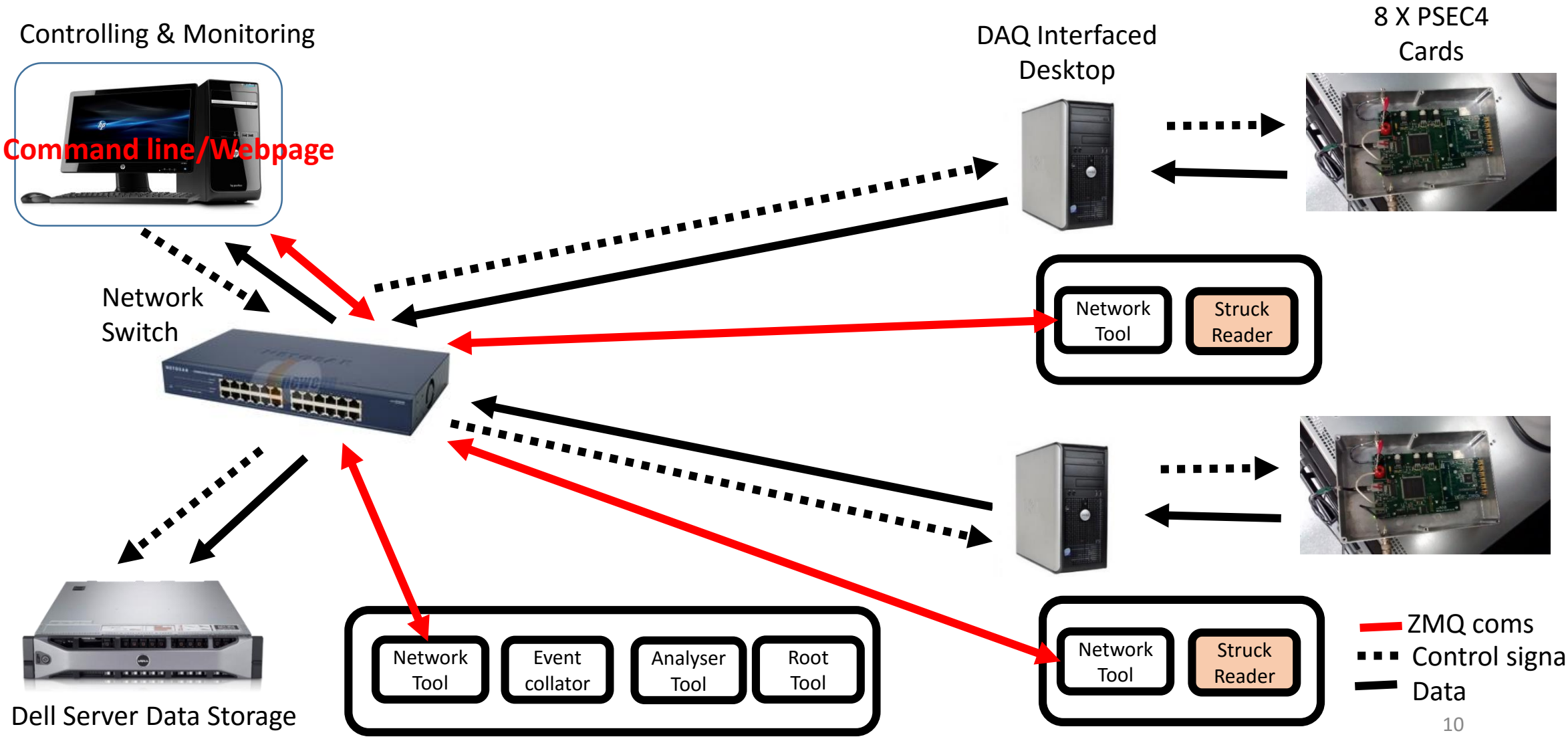
Can add multiple modular board reader Tools or simulated board reader tools to simulate a large detector

72 PMT rings = 72 Crates
576 PSEC4 Cards
3456 PMTs



3) Distributed hardware testing

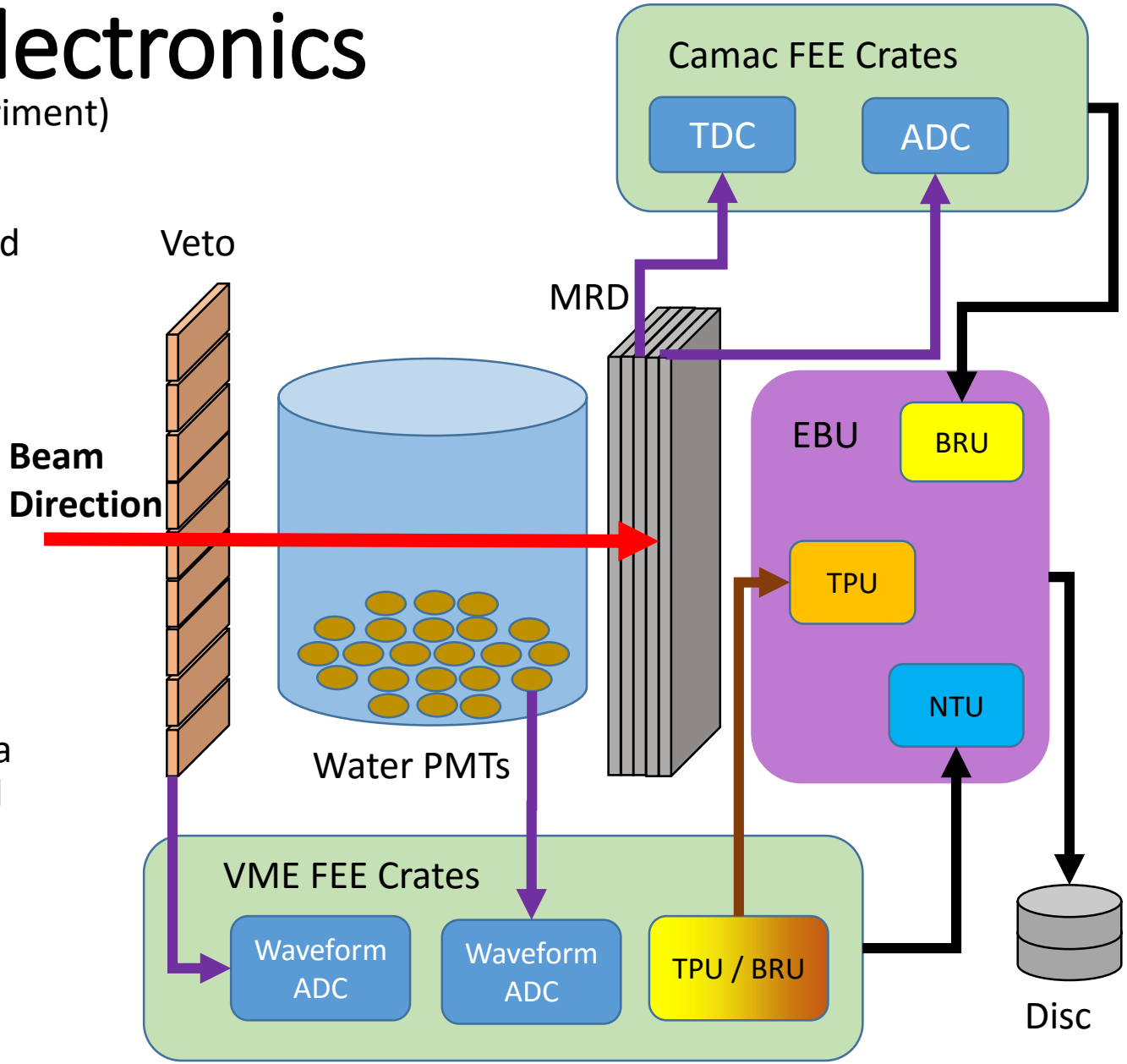
Due to the built in **dynamic service discovery** and the native expandability of the **ZMQ** 1-N sockets
This allows for easy distributed computation



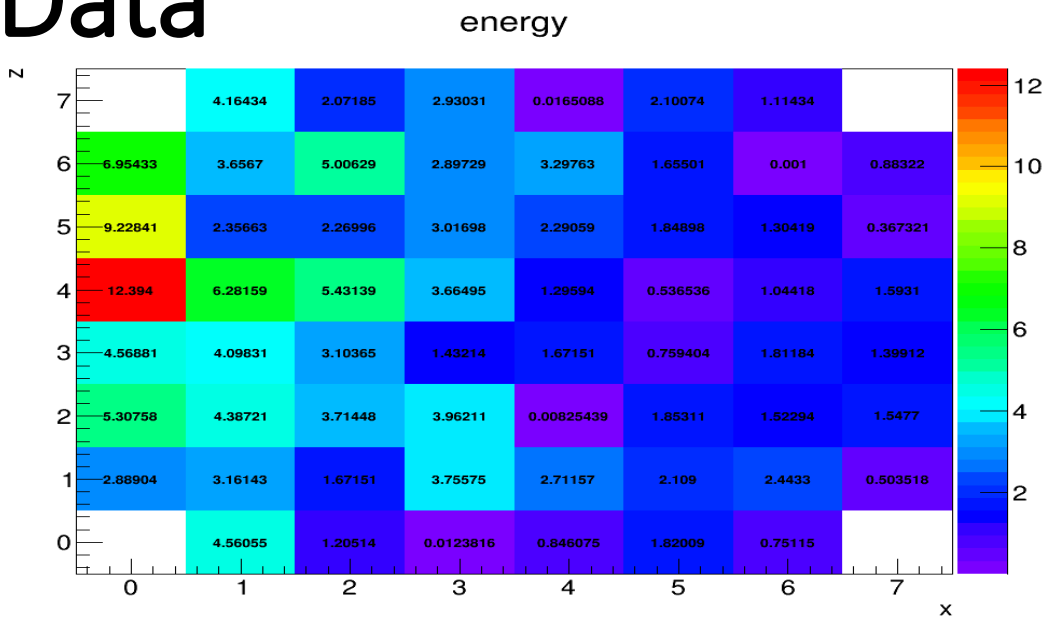
ANNIE Detector and Electronics

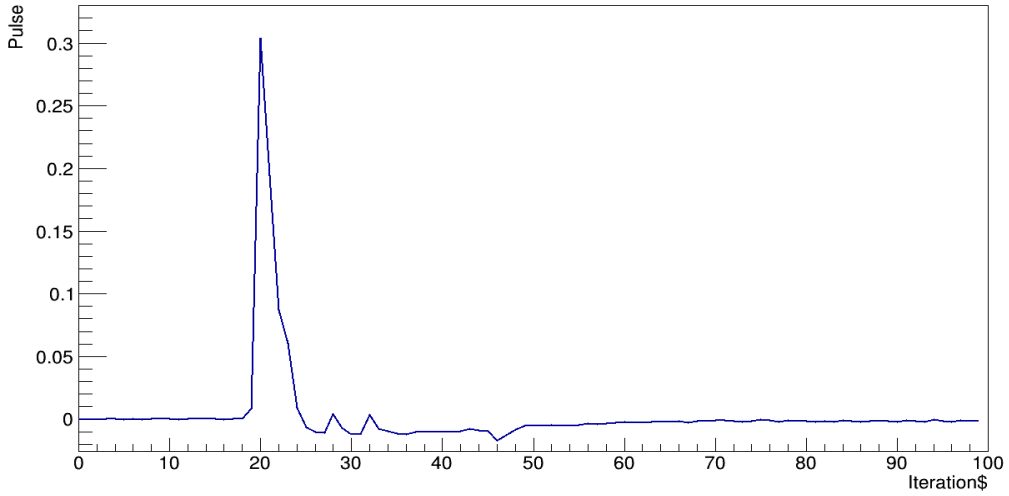
(Atmospheric Neutrino Neutron Interaction Experiment)

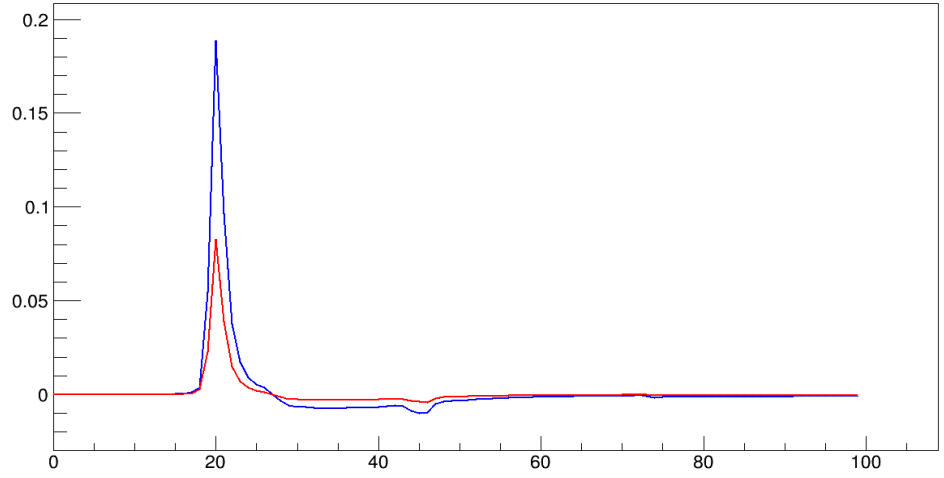
- The ANNIE experiment is a prototype neutrino detector currently taking data Fermilab USA. It aims to be a test bed for many new technologies from Gd-doped water interaction of neutrons to the use of early prototype LAPPDs (Large Area Picosecond Photodetectors).
- The UK Hyper-K group have used this opportunity to undertake R&D and testing of DAQ software and tools for future use in the Hyper-K experiment.
- The use of ToolDAQ in ANNIE has allowed for both real world deployment and testing of this software, as well as aided in its design and continued development.
- The experiment has multiple simultaneous data sources (a forward veto, water based Cherenkov detecting PMTs and a MRD Muon range detector) as well as a blend of front end electronics technologies (VME, Camac and custom FADCs) for ADC/TDC and waveform digitisation, which have allowed extensive testing of the flexibility of the software and all the above features to take place within a single deployment



ANNIE Phase 1 Is Taking Data

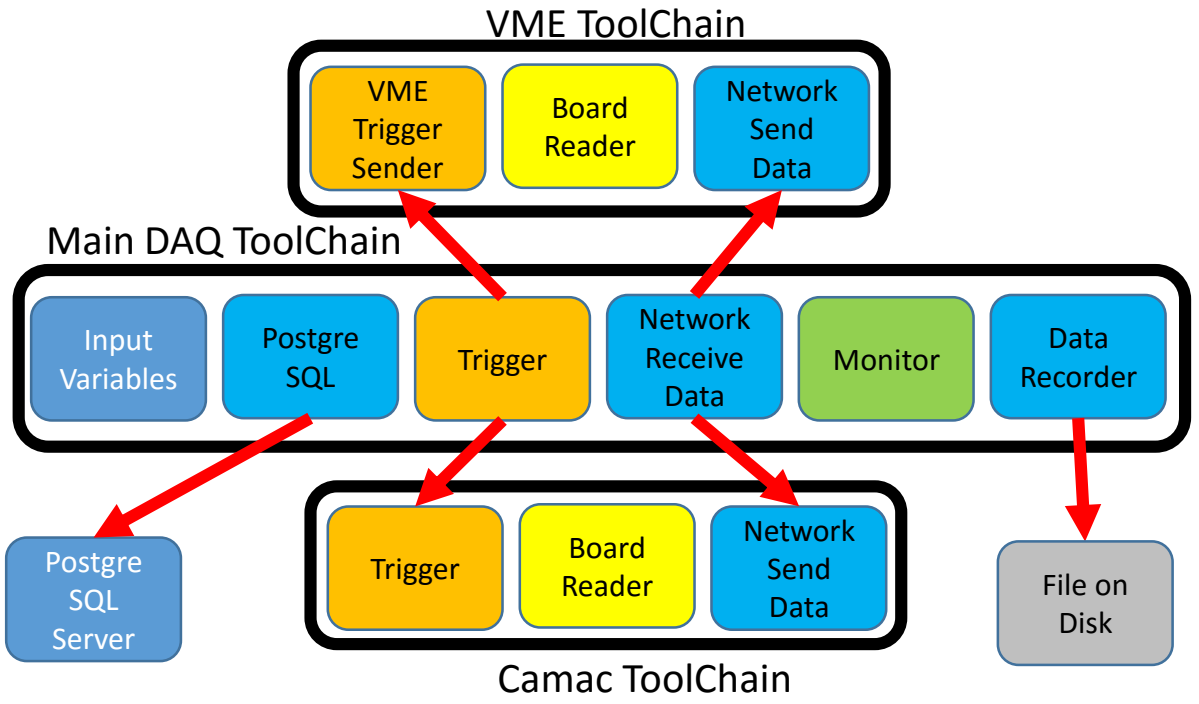


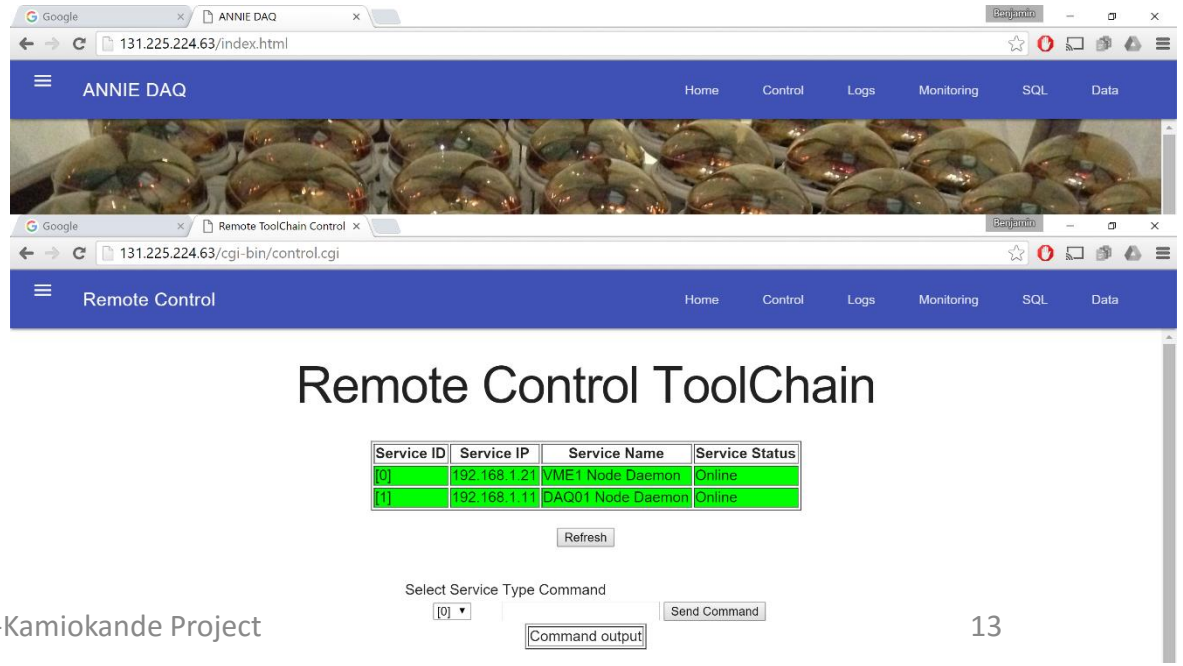




ANNIE's ToolDAQ

- ToolDAQ Is used to control all aspects of the ANNIE experiment. (Water PMTS, MRD, Veto, HV, triggering, run control, data processing, monitoring and databases)
- It is run on many nodes with different OS from rack mounted servers to FEE CPU cards to read out the VME and CMAC crates. Each is comprised of ToolChains of modular units.
- These Crate nodes transmit data over network infrastructure, are responsible for trigger polling of both the ADCs and TDCs and configuring the trigger types to the MT boards and some formatting of output data.
- The more central server node is responsible for passing user configurations to the direct hardware chains coordinating triggering and collating output data over the network for short and persistent storage.
- It also contains real time online monitoring, direct and remote user control, database control and forms of post processing. Other pieces of software such as Web interfaces and remote control systems have been developed to interface with ToolDAQ



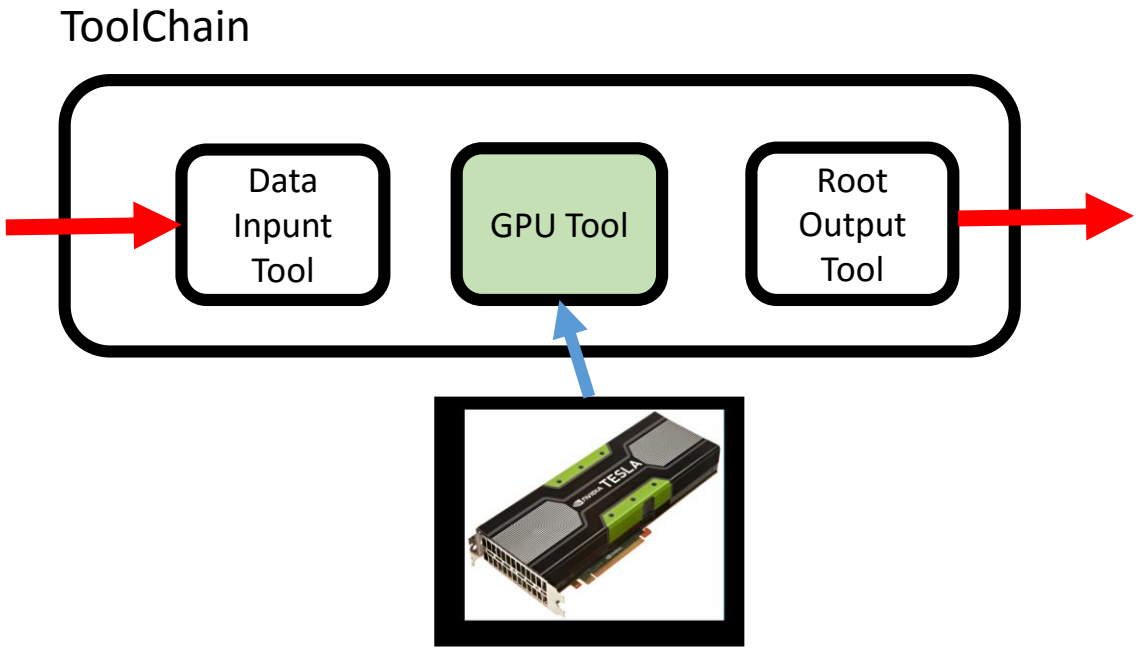


ToolDAQ GPU TPU Development

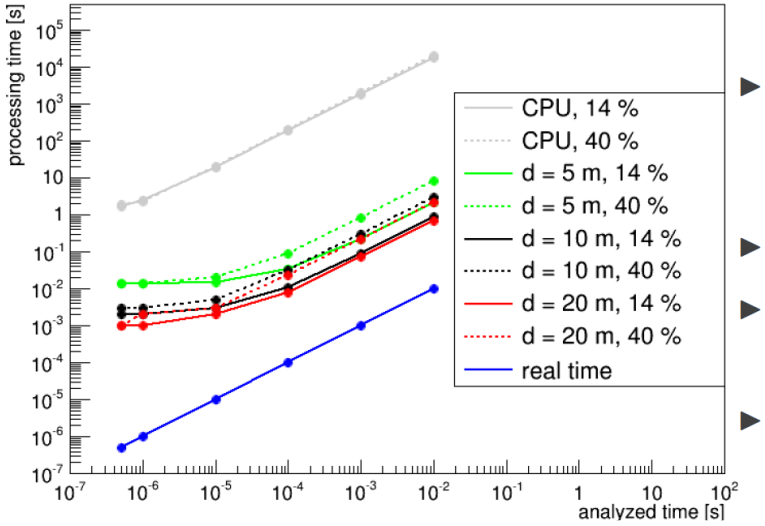
ToolDAQ has already been successfully used in test bed deployments for HyperK GPU trigger algorithm integration.

- RAL rack mounted test stand
- Emerald GPU cluster

Results have show 4 orders of magnitude decrease in processing time.



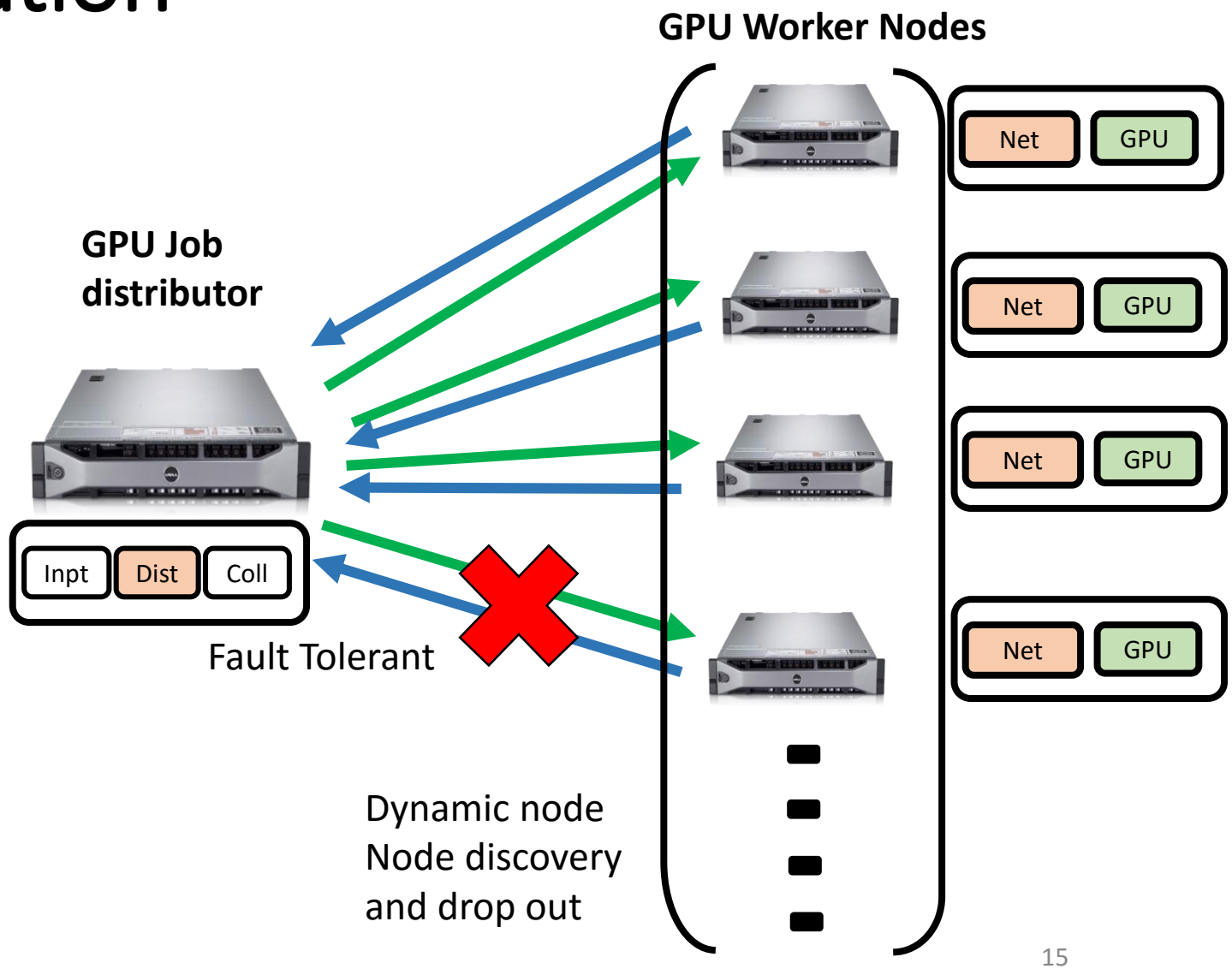
Test Vertices Algorithm Processing time



- GPU reduces the time with respect to CPU by 4 orders of magnitude
- ideal data chunk $= 1$ ms
- need 400 GPUs for real-time processing
- projected cost $\approx 400k\$$

ToolDAQ GPU development: Fault tolerant distributed computation

- GPU trigger development has been expanded to create the DAQ infrastructure to undergo distributed TPU computing in ToolDAQ
- Fault tolerant node communication has been developed to allow for loss of communication due to hardware or network infrastructure failure
- Intelligent job distribution allows no trigger data to be lost by rerouting to available nodes
- Dynamic service discovery and job distribution allows for nodes hardware to be added and removed on the fly, without hardcoding or restarting the system.



Hyper K Benefits

Despite being used elsewhere ToolDAQ has always been designed to meet HyperK's requirements.

- A lot of DAQ experience, knowledge and R&D has been achieved with ANNIE experiment.
- ToolDAQ has seen many improvements with real world deployment that will be directly usable by Hyper K (DAQ control, fault tolerance, expanded feature set,etc.)
- Any code developed for ANNIE modules can cross over / be used by Hyper K. Hyper K design uses similar block component (FEE BRU EBU TPU) structure so network coms experience, persistification, monitoring, database hosting, Web control interfaces.
- ToolDAQ has already been used in test bed deployments for HyperK GPU trigger algorithm integration into a DAQ frame work, as well as other hardware testing.

HK Reference Design