g – 2’s Analysis Computing

James Mott

g – 2 / SCD Workshop - 03/02/22

Introduction

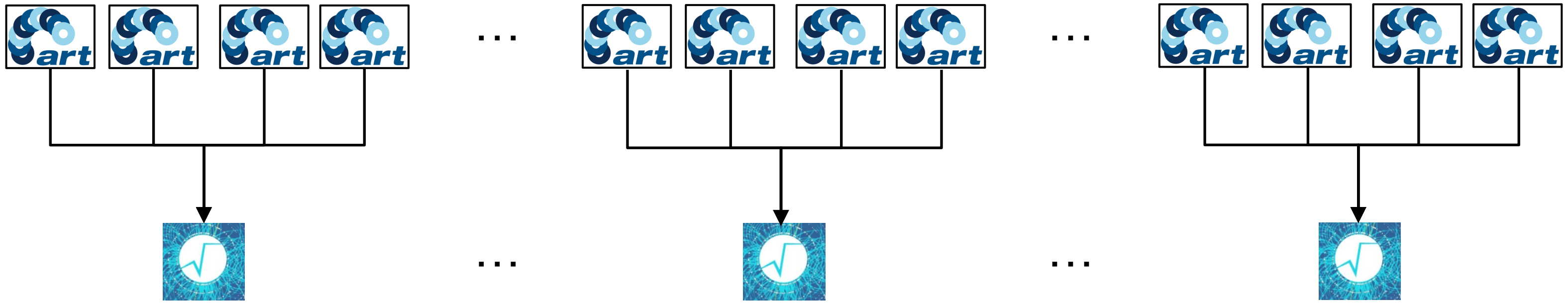
- Aim: sketch out what we use for our analysis processes
- There are many different analyzers, each with their own workflow – I won’t capture all that detail
- I will try and give a representative picture of what roughly happens
- We have separate analysis programs for magnetic field and “fast” detectors that see $\mu^+ \rightarrow e^+$ decays
  - Focus on e^+ data as this is more relevant for computing resources

A Typical e^+ Analysis Workflow

- Start with ***art* files** from production:

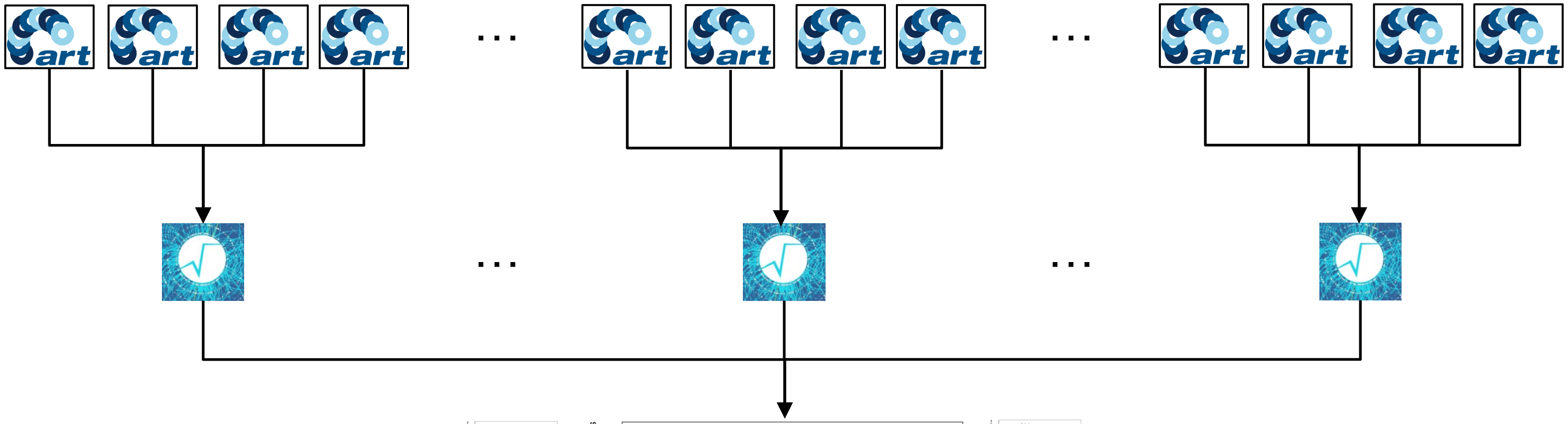
- Usually **process 1 "dataset" at a time**, defined by running conditions
  - Vary in size, but typically might have **35k files & 70 TB**
  - Slimmed production files can reduce this to 10 - 20% in size
  - Add >1 datasets together at end, but need to check consistency first
- Whole process is **repeated many times**:
  - Run-2/3 analysis: **~20 datasets**.
  - Run-4/5: **2 – 3 times as much data**
  - Will need to run multiple times for some systematic scans

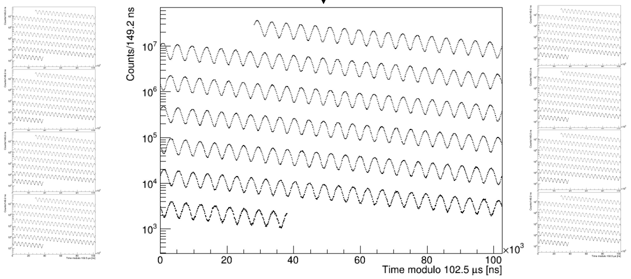
A Typical e^+ Analysis Workflow



- art::analyzers used to create **histograms or trees** in ROOT files
- Usually **perform simple operations** like histogram filling, time-randomization, searching for nearby calo hits for pile-up estimation etc.
  - Some analyzers find it **hard to maintain high CPU efficiency** due to **large fraction of i/o** compared to computation
- Hard to keep **under 2 GB** memory target:
  - Basic unit is 5000 time bins x 200 energy bins x 24 calorimeters
  - Some analyzers do get under 2 GB, others ask for more slots
- Typically takes a few hours (< 12h) to run jobs + recovery

A Typical e$^+$ Analysis Workflow



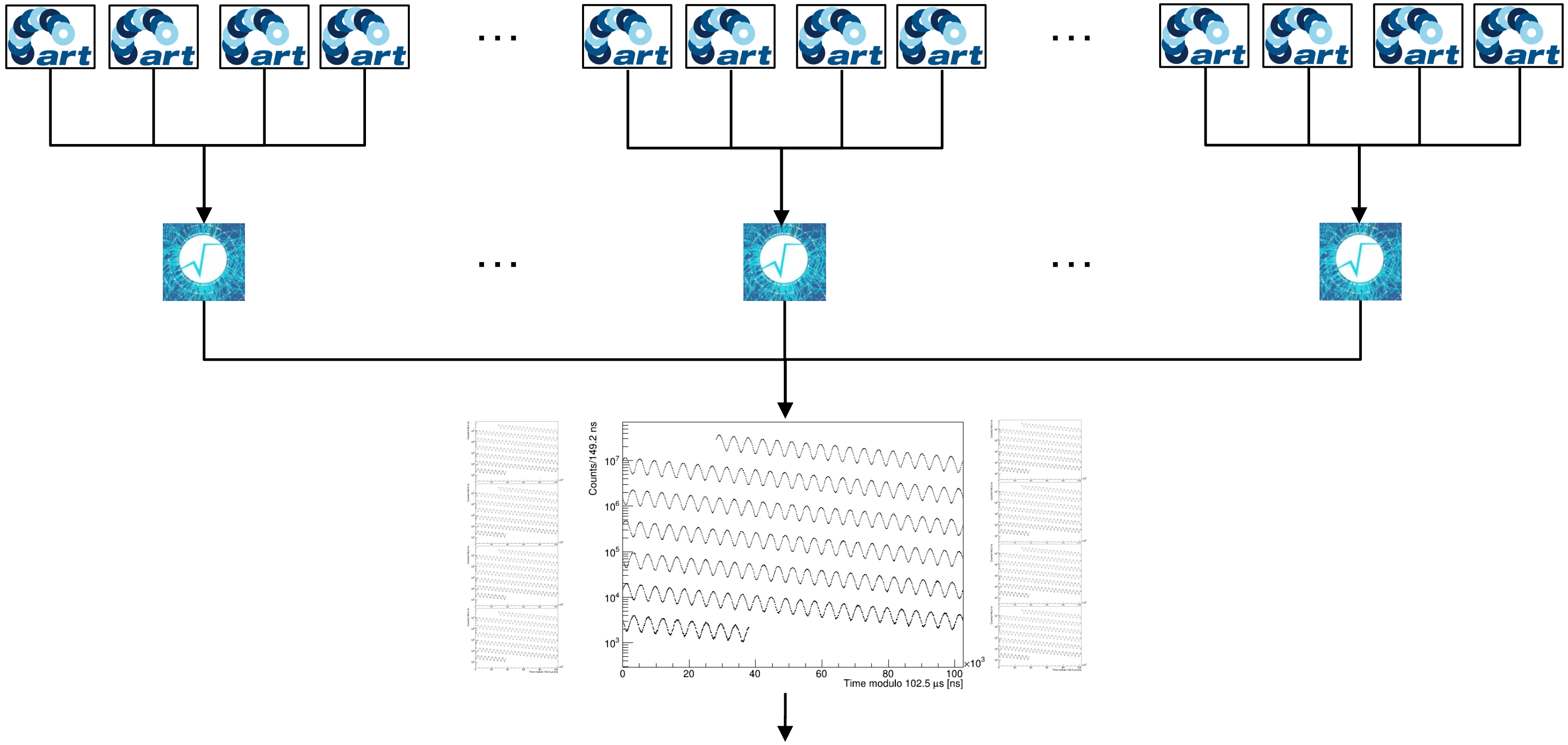


- Create file with **histograms for final stage** of analysis

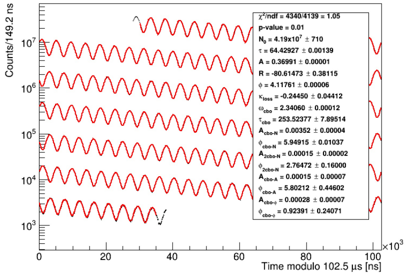
- **Histogram Input**: hadding many files (usually run locally – takes ~1 – 2 h)

- **Tree Input**: Fill histograms (grid or local – slower but with more flexibility)

A Typical e^{+} Analysis Workflow

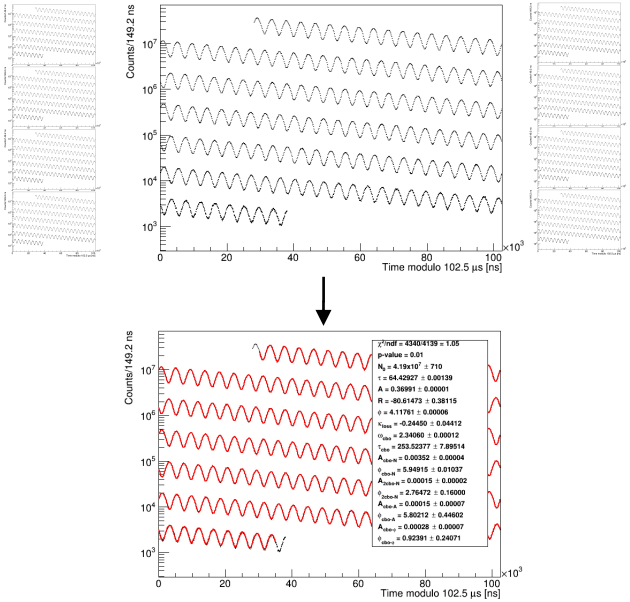


- **Fit preparation**: subtract pile-up, weight data, collapse TH2s to TH1s…



- **Do the fit**: extract parameters of interest

Fitting & Analysis Frameworks



- We placed **high value on independent analyses** to ensure robustness
- Means **every analyzer** has their **own code base** – I'd guess we have ~15 at one time
- Most use **C++ and ROOT.**
- Few use **python**: both with & without ROOT

- Some code is in **main redmine repositories**, and some is **separate**
- At least one uses **fhicl** for configuration settings.
- **Magnetic field** analysis has separate DAQ and workflow.
  - Smaller scale: production takes a few days and then analysis proceeds on single server

Default User Interface

- **4 x virtual machines** (gm2gpvm01,02,03,04)
  - 4 single-core CPUs, 12G RAM, minimal local storage

- **Worked well for early stages** of analysis: mostly development, smaller datasets, fewer users…

- Each machine is **easily overloaded** (single users kill machine by mistake) & **building is very slow**

- Doesn't match well to a lot of our current needs:
  - **Fitting can be slow** (~1-2 m per fit) but is parallelizable: 4 separate machines with 4 cores doesn't make that easy
  - **Large trees as input** don't work well (network speed can be slow)
  - Similarly for **hadding** large numbers of histograms

- Needed something to **fill gap** between the grid and virtual machines

Other User Interfaces

- Several groups have bought own machines that are supported by SCD:

gm2cornellsrv1
32 CPU
250G RAM
300 TB HDD

gm2fnalsrv1
48 CPU; 1 GPU
192G RAM
7 TB SSD

g2field-server
4 CPU; 1 GPU
32G RAM
50 T HDD; 14 T SSD

gm2ucl, gm2liv, gm2ita…
~40 CPUs,
3 – 30 TB HDD

- Typically used for
  - **Parallelization**: fitting, toy MC, building
  - **Filling trees** into histograms
  - Memory intensive tasks (histogramming, 6D acceptance studies)
  - GPU for ML training or frequency extraction (field)

- This **model works fine** for us now but there are some issues:
  - **Not a universal resource** for all collaborators – some are running on personal machines or on clusters at their institutes
  - GPUs are generally underutilized