XRootd as a web server for ROOT-Eve and more

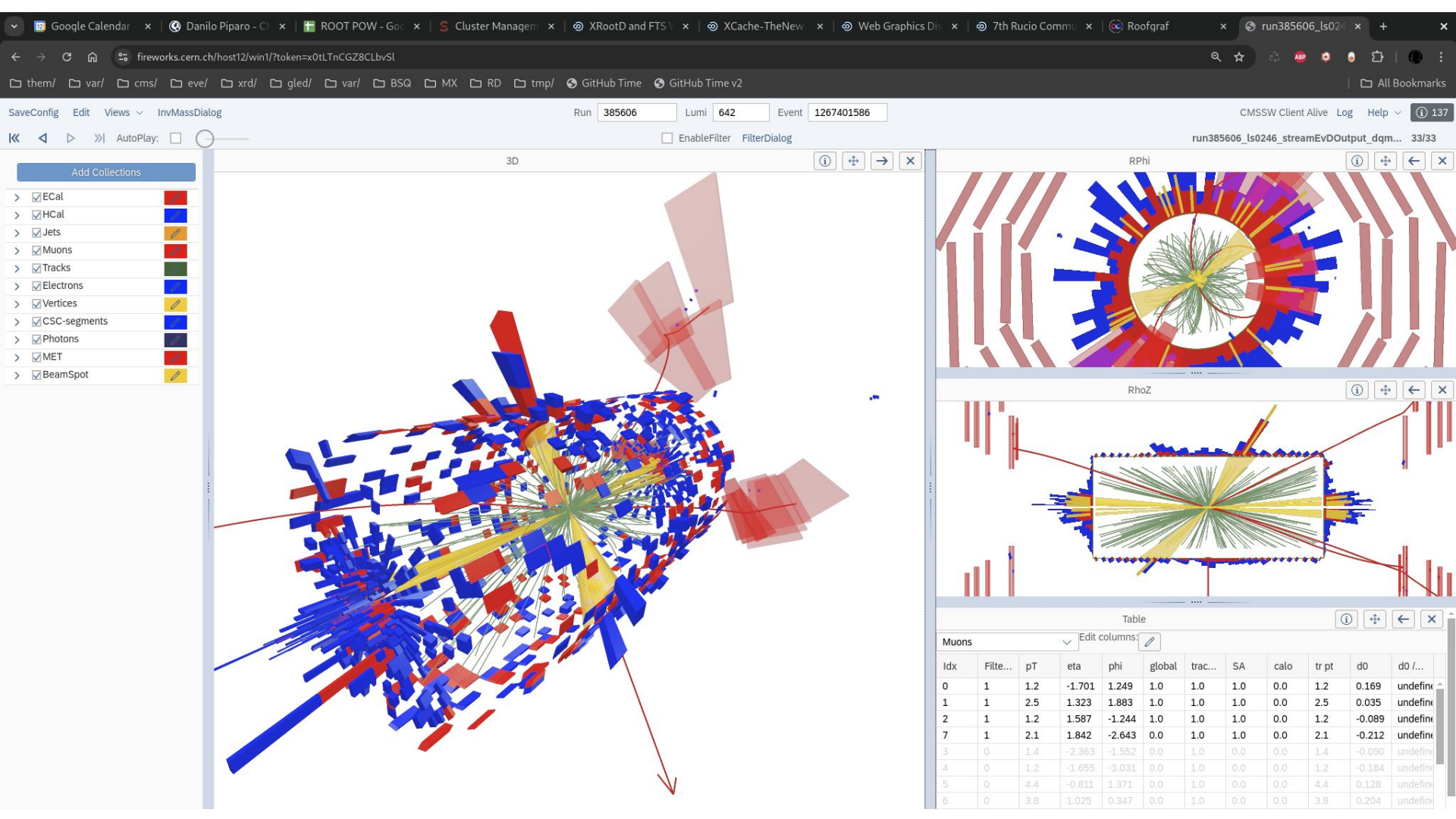
Disclaimer: super preliminary, early idea stage

XRootD Workshop @ SFTC UK, Abingdon

September 13, 2024 Matevž Tadel, UCSD

Initial motivation – introduction

- My other night job is ROOT-Eve and CMS Fireworks
  - physics analysis oriented event visualization (EVE ~ Event Visualization Environment)
- Transition to web-based client-server over the last couple of years
  - Client: JavaScript – JSRoot, OpenUI5, RenderCore
  - Server: ROOT – THttpServer, RWebWindow
    - uses wrapped up CivetWeb – embeddable web server with SSL support
    - The new web-based ROOT GUI also uses this (THREE.js instead of RenderCore).
- FireworksWeb – Event Display as a Service
  - CMS members can access dedicated servers at CERN & UCSD
  - Access data from eoscms, AAA (through XCache), CERNBox *(share with cms-vis-access)*
  - Proto-app with preloaded data-formats forks off an instance (fast!, can serve multiple users/tabs)
- Setting this up at CERN was a major PITA
  - Apache frontend with SSO & OAuth (good, also at UCSD), full proxying through Apache
    - redirect to instance (still proxied), support upgrade to WebSockets
    - oh, and you need Let's Encrypt certs (well, you need those anyway, unless you hate your users)


Google Calendar
Danilo Piparo - Cl
ROOT POW - Goo
Cluster Managem
XRootD and FTS
XCache-TheNew
Web Graphics Dis
7th Rucio Commu
Roofgraf
run385606_ls024
fireworks.cern.ch/host12/win1/?token=x0tLTnCGZ8CLbvSl
them/
var/
cms/
eve/
xrd/
gled/
var/
BSQ
MX
RD
tmp/
GitHub Time
GitHub Time v2
All Bookmarks
SaveConfig
Edit
Views
InvMassDialog
Run 385606
Lumi 642
Event 1267401586
CMSSW Client Alive
Log
Help
137
AutoPlay:
EnableFilter
FilterDialog
run385606_ls0246_streamEvDOutput_dqm... 33/33
Add Collections
ECal
HCal
Jets
Muons
Tracks
Electrons
Vertices
CSC-segments
Photons
MET
BeamSpot
3D
RPhi
RhoZ
Table
Muons
Edit columns:
Idx Filte... pT eta phi global trac... SA calo tr pt d0 d0 /...
0 1 1.2 -1.701 1.249 1.0 1.0 1.0 0.0 1.2 0.169 undefine
1 1 2.5 1.323 1.883 1.0 1.0 1.0 0.0 2.5 0.035 undefine
2 1 1.2 1.587 -1.244 1.0 1.0 1.0 0.0 1.2 -0.089 undefine
7 1 2.1 1.842 -2.643 0.0 1.0 1.0 0.0 2.1 -0.212 undefine
3 0 1.4 -2.363 -1.552 0.0 1.0 0.0 0.0 1.4 -0.090 undefine
4 0 1.2 -1.655 -3.031 0.0 1.0 0.0 0.0 1.2 -0.184 undefine
5 0 4.4 -0.811 1.371 0.0 1.0 0.0 0.0 4.4 0.128 undefine
6 0 3.8 1.025 0.347 0.0 1.0 0.0 0.0 3.8 0.204 undefine

Who in their right mind would volunteer to deploy such a thing?

Especially at CERN or at FNAL, etc.
But, also … at your home institution, for, e.g., online display.

XrdHttp for REve (and, potentially, other ROOT graphics)

- Easier setup of services:
  - Trusted – they run EOS, don't they? Does not need Apache fronting / proxying
  - Web certs still needed
  - Open ports – or local SOCKS proxy
- Authentication & Authorization
  - Use standard mechanisms we're using anyway
  - XrdCl support for CERN SSO is planned
- "Active directories", e.g., */win1/, /3DView/, /RPhiView/, /MuonTable/, …*
  - Those get upgraded to WebSockets, permanent connection – connect to C++ handlers.
  - REveManager manages connections and knows which connection is connected to each view.
- The rest are served as normal files / directories, e.g.:
  - */ui5/distribution/resources/sap-ui-core.js*
  - */ui5/eve7/rcore/REveRenderCore-min.mjs*
  - */ui5/eve7/sdf-fonts/LiberationSerif-Regular.png*

Other uses for XrdHttp in ROOT

- Data / stream / object server – with some namespace definition
- Inspection of files, filtering, decompression
  - In general, creation of secondary products from ROOT files
  - Note – we have *cling* at this point and can compile filtering expressions
- What are the data sources in this case?
  - Pre-determined – fixed, for specific purpose.
    - E.g., ATLAS magnetic field integrator for high-precision muon refit from analysis
  - Specified at connect time – then kept for the duration of the session.
    - Presumably connect to local source or XCache
    - Filter events.
- Probably use similar philosophy as FireworksWeb service:
  - pre-primed server (libs, dat formats, dicts) that forks on demand

Usage of ROOT in XRootd / XCache

- Now, the other way around – what can we do with ROOT in XRootd?
  Well, more or less the same thing … put it works differently.
  - Handlers, plugins. Object access. Filtering, JIT compilation.
  - Also – cling, build dictionaries for XRootd, call desired XRootd APIs.
- Connect to an XCache directly, open a file, then say, through additional http channel that gets routed into ROOT handler!:
  - I know there is this file open on your cache … could you please run this filter for me and a) pass me the first hundred or so hits; then b) put all of them into some file / response
  - Or, even: For these branches, please preload cache with relevant baskets and send a message to a CE / some job manager that data are ready.
  - Or, even more: also unzip the data and repack the requested events into corresponding Events.
- ROOT handlers get **direct access into cache**, calling low-level XrdPfc::File functions directly.

Cache becomes a data transformation tool!

Discussion …

- Already discussed a bit with Andy & Brian, and Guilherme & Cedric
  - Should, probably, be rather easy to make some first tests
- I hoped to talk to my ROOT colleagues beforehand …
  - but let's see what the X-prefix folks think first.
- How to bootstrap?
  - xroot starting root or the other way around? Both, or either :)
    - Getting root into anything is really rather trivial (unless you want TRint console)
  - Forking?
  - Termination? Heh … the chimera will probably die of natural causes anyway …
- Deployment?
  - ROOT can build XRootd – *builtin_xrootd*
  - Most VO stacks already build or have both as externals