



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



Istituto Nazionale di Fisica Nucleare

Attività UniBO per esplorazione analisi interattiva

Tommaso Diotallevi

WP5 Meeting 22/03/2023

Analysis Facility reference

Wiki: <https://inf-n-cms-analysisfacility.readthedocs.io/en/latest/>

Grazie a tutto il Dev team, in particolare Diego Ciangottini e Tommaso Tedeschi, per il supporto ed il debug super reattivo!!



Use case utilizzato

Analisi dati, in corso, della Collaborazione CMS: “*Heavy Neutral Lepton (HNL) search in D_S decays*”.



Lavoro portato avanti da **Leonardo Lunerti**, PhD dell’Università di Bologna, che ringrazio per la disponibilità e per avermi fornito il codice in maniera riproducibile e sequenziale!

Dataset utilizzato:

B-parking dataset, contenente eventi RAW salvati immediatamente su Tape storage, durante la fase di High-Level Trigger (HLT). Attraverso questa metodologia è possibile salvare eventi con rate maggiori rispetto a quelli di trigger, in quanto non viene fatta una ricostruzione prompt.



Enorme mole di dati in input.

Questa analisi nasce in PyROOT, con adozione dalle origini di RDataFrame.



Workflow dell'analisi

1. Data skimming e preprocessing

Input: Formato "heavy" (MiniAOD), proveniente dal dataset B-Parking. \longrightarrow **Size:** \approx 700TB (Data)

Output: flat ntuple, momentaneamente salvate al Tier-2 LNL. \longrightarrow **Size:** \approx 0.5TB (Data) - <1TB (Data+MC)

2. Selezione del candidato HNL migliore

Input: Output step 1.

Output: Flat ntuple, più leggere di quelle in input.

3. Analisi e calcolo limiti

Input: Output step 2.

Output: Risultati fisici (istogrammi, limiti, ...)



Workflow dell'analisi

- CMS Distributed Analysis (CRAB)
- Interactive Analysis (Analysis Facility)

1. Data skimming e preprocessing

Tempo di esecuzione: \approx 2-3 giorni (una tantum)

Input: Formato "heavy" (MiniAOD), proveniente dal dataset B-Parking. \longrightarrow **Size:** \approx 700TB (Data)

Output: flat ntuple, momentaneamente salvate al Tier-2 LNL. \longrightarrow **Size:** \approx 0.5TB (Data) - <1TB (Data+MC)

2. Selezione del candidato HNL migliore

Tempo di esecuzione 2.+ 3. pre-AF \approx qualche ora

Input: Output step 1.

Output: Flat ntuple, più leggere di quelle in input.

3. Analisi e calcolo limiti

Input: Output step 2.

Output: Risultati fisici (istogrammi, limiti, ...)

Step considerati per
use-case Analysis
Facility*



* Workflow dell'analisi ad ora non definitivo. Possibile futura eliminazione di step 2, con l'utilizzo di tutti i candidati HNL.

Input/Output

L'utilizzo dell'Analysis Facility richiede un porting delle operazioni di IO.

Analisi “originaria”:

- Lettura e scrittura “locali” dell'analisi (Tier-3 Bologna);
- Batch system basato su HTCondor in shared fs (CERN lxplus o Tier-3 Bologna);

Porting su Analysis Facility:

- Lettura e scrittura necessariamente “remoti”:
 - Lettura tramite xrootd AAA;
 - Stage-out tramite Davix (WebDAV/HTTP) al Tier-2 Legnaro. Problema legato alle configurazioni dei vari siti grid, con permessi in scrittura xrootd ora generalmente rimossi, con differenze regionali.

In entrambi i casi, viene fatto book-keeping dei file ad-hoc: JSON file, con elenco dei vari file path. In questo modo, è possibile un passaggio tra i vari step del workflow, descritti in precedenza.



Scenari testati dell'Analysis Facility

- Gli scenari testati, sono principalmente due:
 1. Scenario HTCondor (non troppo distante da uso “classico” per analisti);
 2. Scenario DASK (uso distribuito delle risorse, caso più interessante e goal finale).

Gli stessi scenari sono quelli descritti, nel dettaglio, dalla wiki di riferimento da cui mi sono basato.



Scenario HTCondor

Scenario usato come test, per provare il porting dell'IO e prendere mano con l'interfaccia Jupyterlab dell'Analysis Facility.

Troubleshooting:

- ☑ La storage area fornita, *persistent-storage*, non è più shared. Risolto con la re-implementare dell'IO tra la submit e la execute machine, per eseguire correttamente i job. Esercizio di ripasso su configurazione HTCondor 😊.
- ☑ L'analisi originale utilizzava un ambiente software cvmfs (/cvmfs/sft.cern.ch/lcg/views/dev4/latest/...). Queste tipologie di pacchetti sft non sono disponibili nell'Analysis Facility (non tutta l'area cvmfs è montata). Risolto tramite immagine apptainer/singularity fornita dai devs: /cvmfs/images.dodas.infn.it/registry.hub.docker.com/dodasts/root-in-docker:ubuntu22-kernel-v1.
- ☑ Redirector nazionale xrootd ha mostrato alcuni problemi: xrootd-cms.infn.it → redirector puntato verso la cache dell'AF. Risolto utilizzando il redirector globale cms-xrd-global.cern.ch.
- ☑ Prove e debugging di stage-out. Davix in scrittura dava problemi di TLS handshake: necessaria la cartella con le grid CA. Presente anch'essa in cvmfs. → Alta dipendenza da cvmfs.



Scenario DASK

In questo scenario, l'analisi è interattiva (utilizzando Jupyter notebook) ed il backend utilizza un cluster Dask, con numero di nodi modificabile dall'utente.

Troubleshooting:

- Utilizzando l'immagine apptainer/singularity (inizialmente la stessa dello scenario HTCondor), il cluster dask aveva problemi di creazione dei workers. Risolto lato dev AF, con patch dedicata!
- Problema con IAM CMS durante authn/authz: la migrazione tra VOMS e IAM ha causato un cambiamento indesiderato dello username, che non è più lo user CERN ma un progressivo numerico. Risolto con sync script, lato dev AF.
- Problema con python jupyter kernel: il kernel da immagine singularity non partiva correttamente ("Kernelspec Manager"). Risolto installando apptainer (migrazione da singularity a apptainer).
- Porting dell'analisi HNL su notebook: necessario passare da RDF a RDF Experimental (da ROOT v6.26+).
 - Comunicazione con cluster dask e configurazione dei proxy sui nodi Dask per AAA.
 - Funzionalità di RDF Experimental ancora limitate (vedi prossima slide).



Scenario DASK II

Alcuni metodi di RDF, utilizzati nell'analisi originale, sono stati solo recentemente integrati a RDF Experimental.

In particolare, `df.GetColumnNames()`, disponibile solo da 6.28+ ([release-notes](#))

Troubleshooting:

- Creazione di un'immagine apptainer/singularity che abbia al suo interno ROOT 6.28 compilato e la configurazione base dell'AF (dask, jupyter hub/lab, ...).
Centos7 problematico (ROOT non compila con l'immagine base suggerita nella wiki "dciangot/jlab-rclone:v7".
Utilizzato infine Alma8, correttamente caricato su [dockerhub](#) personale ([repo GitHub con CI](#)).
- Immagine utilizzata per creazione cluster Dask su Analysis Facility e per creazione kernel python mediante il "Kernelspec Manager".
Con l'utilizzo di questa immagine, tuttavia, non è stato ancora possibile creare correttamente un cluster dask, e ad aggiungere nuovi workers (problema analogo al troubleshooting descritto nella slide precedente).
[Debugging in corso!!](#)

A che punto sono?

- Esecuzione corretta del workflow dell'analisi originale su HTCondor dell'Analysis Facility. Per questioni di semplicità, il workflow è stato girato per ora solo su un segnale Monte Carlo piuttosto leggero.
- L'esecuzione su Dask del workflow dell'analisi originale è quasi completo: ultimi debug in corso, per problemi con immagine custom (ROOT 6.28) —> vedi slide precedente
 - Possibile problema futuro: altre mancanze di RDF Experimental che potrebbero bloccare l'esecuzione del codice originale. Nel caso: trovare workaround nell'analisi o richiedere funzionalità RDF Experimental.

Prossimi passi

- Scalare su tutti i dati/MC dell'analisi. Suggerimenti su possibili bottleneck (IO AAA, CPU,...);
- Monitoring: una volta “pronti”, sarà necessario valutare delle metriche ben precise (timing, uso di risorse,...)
 - ▶ Esistono dashboard Analysis Facility? Kibana/Graphana?



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Grazie per l'attenzione

tommaso.diotalevi@unibo.it

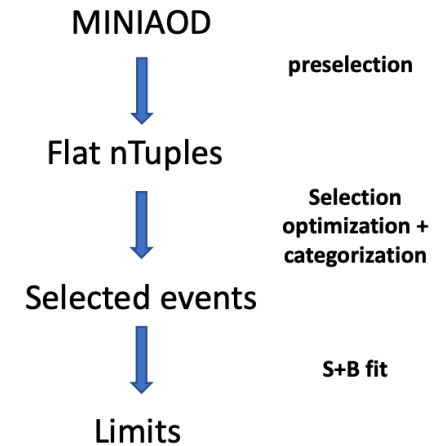
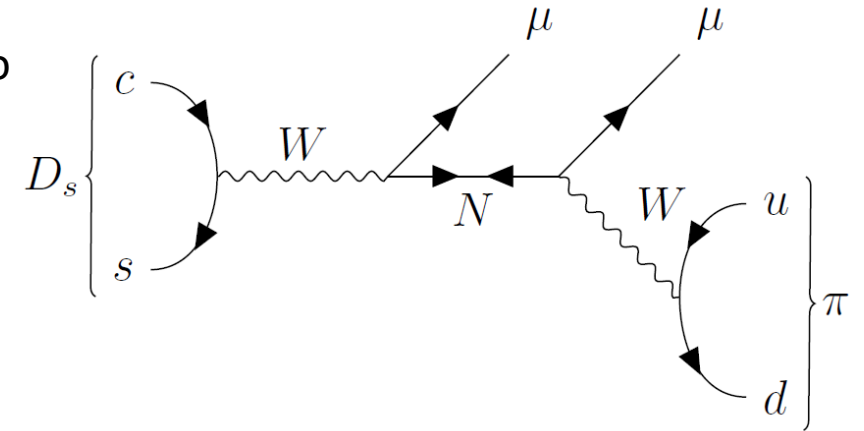
Analisi Heavy Neutral Lepton (HNL)

Ricerca di un leptone pesante neutro N , proveniente dal mesone D_S , nello stato finale contenente un μ e un π .

Alla firma sperimentale viene inoltre considerato un μ aggiuntivo, proveniente dal W iniziale. I due muoni possono avere sia stesso-segno che segno-opposto.

Strategia:

- Analisi sul dataset intero B-Parking Ultra-Legacy re-reco.
- Discriminazione segnale-fondo:
 - Campioni QCD μ -enriched, per modellare la shape del fondo;
 - Campioni di segnale per $m_{HNL} = 1.0, 1.5$ GeV e $c\tau_{HNL} = 10, 100, 1000$ mm.
- Stima del background è data-driven: per ogni ipotesi di massa del neutrino.
- Stima del segnale è data dal canale di normalizzazione: $D_S \rightarrow \phi(\rightarrow \mu\mu)\pi$



Talk di Leonardo Lunerti al working meeting B-Physics: [link](#) (CMS restricted)