

VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE TEHNOLOGIJE

ZAVRŠNI RAD NA MASTER STRUKOVNIM STUDIJAMA

Replikacija analize podataka za Higsov bozon: Raspad u četiri leptona u ROOT Framework-u

Mentor:

dr Aleksandar Simović

Student:

Viktor Varkulja 469/23

Beograd, 2025

VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE TEHNOLOGIJE



ZAVRŠNI RAD NA MASTER STRUKOVNIM STUDIJAMA

Replikacija analize podataka za Higsov bozon: Raspad u četiri leptona u ROOT Framework-u

Mentor:
dr Aleksandar Simović

Student:
Viktor Varkulja 469/23

Beograd, 2025

SAŽETAK

Predmet ovog master rada je replikacija analize raspada Higsovog bozona u četiri leptona ($H \rightarrow ZZ \rightarrow 4l$) korišćenjem podataka dostupnih preko CERN Open Data portala. Cilj istraživanja bio je da se ispita u kojoj meri je moguće ponoviti kompleksne analize iz oblasti fizike visokih energija u nezavisnom istraživačkom okruženju, uz ograničene tehničke resurse.

Metodologija je obuhvatila više nivoa replikacije, koji su postepeno povećavali obim analize: od početnog testiranja koda, preko vizuelne validacije rezultata, do obrade velikih skupova podataka u četvrtom nivou. Za rad su korišćeni CMSSW softverski okvir i ROOT biblioteka, dok je obrada realizovana na Google Cloud Platform infrastrukturi, putem virtuelnih mašina zasnovanih na CernVM image-u. Uz primenu kontekstualnih fajlova i automatizovanih skripti postignuta je visoka reproduktivnost i efikasnost.

Rezultati su pokazali da je, uprkos delimičnoj nedostupnosti Monte Karlo setova podataka, moguće dobiti stabilne histograme i ponoviti očekivani rezultat. Posebno je potvrđeno da stvarni podaci daju rezultate koji su u skladu sa očekivanjima. Zaključeno je da otvoreni podaci predstavljaju pouzdanu osnovu za replikaciju i edukaciju, ali i da buduća istraživanja treba da se usmere na proširivanje dostupnosti podataka i još veću automatizaciju procesa obrade.

Ključne reči: Higsov bozon, CERN Open Data, replikacija analize, ROOT, Google Cloud

ABSTRACT

The subject of this master thesis is the replication of the Higgs boson decay into four leptons ($H \rightarrow ZZ \rightarrow 4l$) using data available through the CERN Open Data portal. The aim of the research was to examine to what extent it is possible to reproduce complex analyses from high-energy physics in an independent research environment, with limited technical resources.

The methodology included several levels of replication, gradually increasing the scope of the analysis: from initial code testing, through visual validation of results, to processing large datasets in the fourth level. The work was carried out using the CMSSW software framework and the ROOT library, while data processing was performed on Google Cloud Platform infrastructure, through virtual machines based on the CernVM image. With the use of contextual files and automated scripts, high reproducibility and efficiency were achieved.

The results showed that, despite the partial unavailability of Monte Carlo datasets, it is possible to obtain stable histograms and reproduce the expected outcome. In particular, it was confirmed that the real data does provide results consistent with expectations. The conclusion emphasizes that open data represent a reliable basis for replication and education, while future research should focus on expanding data availability and the further automation of the processing workflow.

Keywords: Higgs boson, CERN Open Data, replication of analysis, ROOT, Google Cloud



ИЗЈАВА О АКАДЕМСКОЈ ЧЕСТИТОСТИ

Студент/киња: _____

Број индекса: _____

Студент/киња мастер струковних студија: _____

Аутор/ка завршног рада на мастер струковним студијама под називом:

Потписивањем изјављујем:

- да је рад искључиво резултат мог сопственог истраживачког рада;
- да сам рад и мишљења других аутора које сам користио/ла у овом раду назначио/ла или цитирао/ла у складу са Упутством за цитирање и израду пописа литературе;
- да су сви радови и мишљења других аутора наведени у списку литературе/референци који су саставни део овог рада и писани у складу са Упутством за цитирање и израду пописа литературе;
- да сам добио/ла све дозволе за коришћење ауторског дела који се у потпуности/целости уносе у предати рад и да сам то јасно навео/ла;
- да сам свестан/на да је плагијат коришћење туђих радова у било ком облику (као цитата, парафраза, слика, табела, дијаграма, дизајна, планова, фотографија, филма, музике, формула, веб сајтова, компјутерских програма и сл.) без навођења аутора или представљање туђих ауторских дела као мојих, кажњиво по закону (Закон о ауторском и сродним правима, Службени гласник Републике Србије, бр. 104/2009, 99/2011, 119/2012, 29/2016 - одлука УС и 66/2019), као и других закона и одговарајућих аката Високе школе струковних студија за информационе технологије у Београду;
- да сам свестан/на да плагијат укључује и представљање, употребу и дистрибуирање рада предавача или других студената као сопствених;
- да сам свестан/на последица које код доказаног плагијата могу проузроковати на предати завршни рад на мастер струковним студијама и мој статус;
- да је електронска верзија завршног рада на мастер струковним студијама идентична штампаном примерку и пристајем на његово објављивање под условима прописаним актима Високе школе струковних студија за информационе технологије.

Београд, _____

Потпис студента/киње

SADRŽAJ

Sažetak	A
Abstract	A
1. Uvod	1
2. Analitički pristup	3
2.1. Kvantna teorija polja	3
2.2. Standardni model	4
2.3. Higsov bozon	5
2.4. Big Data u fizici čestica	5
2.4.1. Značaj Big Data tehnologija u fizici	6
2.4.2. Tehnologije i upotreba danas	7
3. ROOT Framework	9
3.1. Arhitektura root okvira	10
3.2. ROOT fajlovi	12
3.3. Funkcionalnosti ROOT okvira	13
3.4. Upotreba ROOT framework-a	15
4. Replikacija analize podataka	19
4.1. Način replikacije analize	19
4.2. Prvi nivo replikacije	20
4.3. Drugi nivo replikacije	21
4.4. Treći nivo replikacije	24
4.5. Analiza Python i C++ fajlova	27
5. Četvrti nivo replikacije pomoću paralelizacije na Google Cloud Platformi	35
5.1 Principi paralelizacije i primena u projektu	35
5.2 Google Cloud Platforma	36
5.3 Četvrti nivo replikacije	37
5.3.1 Pregled ulaznih podataka	38
5.3.2 Priprema virtualnih mašina za obradu podataka	40
5.3.3 Rezultati obrade i vremenska analiza	45
6. Zaključak	52
Reference	54
Prilog	55
Popis skraćenica	55
Popis slika	56
Popis grafika	56
Popis listinga	56
Popis formule	57
Radna Biografija (CV)	57

1. UVOD

Razvoj fizike elementarnih čestica tokom poslednjih decenija oslanja se na sve sofisticiranije eksperimente i kompleksnije analitičke metode. Veliki hadronski sudarač (LHC – Large Hadron Collider) u CERN-u (Organisation européenne pour la recherche nucléaire – Evropska organizacija za nuklearno istraživanje) predstavlja centralnu tačku ovih istraživanja, gde se na najvišem energetskom nivou testiraju predviđanja Standardnog modela i ispituju potencijalni tragovi nove fizike. Među najznačajnijim rezultatima koji su proizašli iz LHC eksperimenata nalazi se otkriće Higsovog bozona 2012. godine, čime je potvrđen ključni element mehanizma koji omogućava česticama da imaju masu. To otkriće nije samo potvrdilo teorijski okvir Standardnog modela, već je otvorilo i nova pitanja u vezi sa stabilnošću tog modela i potencijalnim pravcima njegovog proširenja [1].

U savremenoj nauci, posebno u oblasti eksperimentalne fizike, sve veću ulogu imaju principi otvorene nauke. CERN je jedan od pionira u implementaciji koncepta otvorenih podataka, omogućivši istraživačima širom sveta pristup rezultatima putem CERN Open Data portala. Dostupni su ne samo sirovi podaci prikupljeni tokom rada detektora CMS (Compact Muon Solenoid) i ATLAS (A Toroidal LHC Apparatus), već i kompletan softverski okvir, dokumentacija i primeri konfiguracija.

Sam Veliki hadronski sudarač (LHC) funkcioniše tako što ubrzava protone u dva snopa koji se kreću u suprotnim pravcima kroz kružni akceleratori tunel dužine 27 kilometara, koristeći snažna supravodljiva magnetska polja za njihovo vođenje i fokusiranje. Kada se ti snopovi ukrste u tačkama detektora, dolazi do sudara pri energijama od nekoliko teraelektronvolti, pri čemu se stvara čitav niz elementarnih čestica, uključujući i retke procese poput raspada Higsovog bozona. Detektori kao što su CMS i ATLAS osmišljeni su da beleže te sudare u slojevima: od unutrašnjih tragova naelektrisanih čestica, preko elektromagnetnih i hadronskih kalorimetara, pa sve do spoljnog mionskog sistema. Na taj način se dobijaju višeslojni podaci o svakoj interakciji, koji kasnije mogu da se analiziraju radi identifikacije čestica i rekonstrukcije njihovih osobina [2], [3], [4], [5].

Otvaranjem tih podataka za širu zajednicu, CERN je kreirao jedinstvenu platformu za edukaciju, istraživanje i proveru metodologije. Ona omogućava transparentnost i reproduktivnost, jer bilo koji istraživač može da pristupi istim podacima kao i originalne kolaboracije, ali istovremeno i podstiče razvoj novih ideja kroz reinterpetaciju i dodatne analize. Ovaj pristup čini da LHC eksperimenti ne ostanu zatvoreni samo u okvirima velikih kolaboracija, već postaju resurs za celokupnu naučnu i obrazovnu zajednicu.

U ovom radu poseban akcenat stavljen je na replikaciju analize raspada Higsovog bozona u četiri leptona ($H \rightarrow ZZ \rightarrow 4l$), koja je jedna od ključnih kanala korišćenih u otkriću. Replikacija se odvija kroz više nivoa koji progresivno povećavaju obim i kompleksnost analize. Prvi nivoi fokusirani su na validaciju koda i testiranje manjih datasetova, dok poslednji, četvrti nivo, obuhvata obradu velikih datasetova koji sadrže desetine miliona događaja. Posebno u četvrtom nivou postaje jasno koliko su tehnički resursi i infrastruktura važni za sprovođenje jedne ovakve analize.

Za realizaciju ove replikacije korišćen je CMS softverski okvir (CMSSW - CMS Software), zasnovan na C++ i Python alatima za obradu i analizu podataka. Ključnu ulogu ima ROOT sistem, koji omogućava rad sa histogramima i kompleksnim datasetovima. Pored softverskog aspekta, važan deo rada odnosi se na infrastrukturu. Korišćena je Google Cloud Platforma (GCP), gde su pokretane virtuelne mašine bazirane na CernVM image-u. Pomoću kontekstualnih fajlova, koji se prosleđuju instancama prilikom njihovog podizanja, automatski se konfiguriše okruženje i preuzima kod za analizu. Time se postiže potpuna uniformnost među svim instancama i obezbeđuje mogućnost skaliranja na desetine mašina bez dodatnog manuelnog rada.

Distribuirana priroda obrade zahteva pažljivo planiranje. Na osnovu merenja iz trećeg nivoa replikacije, gde je prosečno vreme obrade jednog događaja iznosilo 0,008732 sekunde, izračunato je da jedna virtuelna mašina u neprekidnom radu tokom 24 sata može da obradi

oko 9,9 miliona događaja. Ova računica poslužila je za određivanje potrebnog broja instanci: za kompletan obim podataka bilo je potrebno pokrenuti ukupno 38 virtuelnih mašina, od čega 23 za stvarne podatke i 15 za MC (Monte Carlo) simulacije. Na taj način pokazano je da analiza ovog tipa zahteva ozbiljnu infrastrukturu, ali i da se uz pažljivo planiranje može izvesti u realnim uslovima ograničenih resursa.

Jedan od izazova sa kojima se susreće analiza jeste nepotpuna dostupnost MC podataka. Na zvaničnom CERN Open Data portalu jasno stoji napomena da su za određene datasetove javno dostupni samo delimični skupovi fajlova. Preuzimanje kompletnog skupa moguće je, ali traje nedeljama ili mesecima. Ovo ograničenje imalo je direktan uticaj na rezultate u četvrtom nivou, jer se u histogramima dobijenim isključivo iz MC podataka jasno vide praznine i statističke nestabilnosti. Da bi se prevazišao ovaj problem, u radu su korišćeni i rezultati prethodnih nivoa replikacije kao dopuna, čime je postignuta stabilnost i konzistentnost konačnih histograma.

Važan aspekt rada odnosi se i na transparentnost metodologije. Svaki korak – od pripreme indexfile-ova, preko dodele datasetova pojedinačnim instancama, do objedinjavanja ROOT fajlova i analize histograma – detaljno je dokumentovan i izveden pomoću jasno definisanih skripti. Na taj način se obezbeđuje potpuna reproduktivnost, što je osnovni princip otvorene nauke. Takođe, prikazani su i svi tehnički izazovi na koje se nailazilo, uključujući prekoračenje očekivanog vremena obrade, rad sa ograničenim budžetom, ograničenost broja instanci po projektu i specifičnosti mrežne konfiguracije (NAT (Network Address Translation) gateway, IAP (Identity-Aware Proxy) tunel), što sve zajedno daje realističnu sliku složenosti ovog tipa istraživanja.

Cilj ovog rada, stoga, nije samo u tome da se dobije histogram mase četiri leptona koji potvrđuje postojanje Higsovog bozona. Podjednako važan cilj jeste demonstracija da je moguće izvesti ovakvu analizu na osnovu javno dostupnih podataka i sa ograničenim tehničkim resursima. Time se pokazuje da koncept otvorenih podataka nije samo deklarativna inicijativa, već praktičan okvir u kojem se složene naučne analize mogu ponoviti i dalje razvijati. Ovaj rad tako spaja naučnu metodologiju sa modernim tehnologijama distribuiranog računanja i nudi primer kako se istraživanja u oblasti visokoenergetske fizike mogu učiniti dostupnim i reproduktivnim širom naučne zajednice.

Rad je organizovan tako da najpre predstavi teoretski okvir i značaj replikacije u kontekstu otvorene nauke, zatim opis korišćenih datasetova i tehničku realizaciju putem Google Cloud infrastrukture, a potom detaljno prikaže postupak analize i diskusiju rezultata. Na kraju se izvodi zaključak o dometima i ograničenjima sprovedenog pristupa, uz napomenu da se metod može dalje razvijati i primenjivati i na druge kanale i datasetove u okviru CERN Open Data portala.

2. ANALITIČKI PRISTUP

Analitički pristup u fizici se bazira na podelama definisanim veličinom i brzinom kretanja objekata.

Klasična fizika: U slučaju da su objekti veliki i sporo se kreću, reč je o klasičnoj fizici. Klasična fizika proučava karakteristike i ponašanje velikih objekata. Gravitaciona sila je glavna sila koja utiče na njih, a Isak Njuton je najbolje definisao suštinu njihovog ponašanja kroz svoja tri zakona mehanike. Međutim, na atomskom nivou klasična fizika nije dobar okvir za razumevanje ponašanja objekata [6], [7].

Glavno ograničenje je da je konzervacija mase neophodna da bi ovaj okvir funkcionisao, ali se često događa da se u nekim interakcijama na atomskom nivou gubi masa objekta. Zato se za objašnjenje ovih prirodnih fenomena uvodi relativizam i relativistička fizika.

Relativistička fizika: Relativistička fizika se bavi sa objektima koji se brzo kreću, otprilike brzinom svetlosti (c). U tom okviru je ideja da se energija i impuls konzervuje. Centralna ideja je specijalne teorije relativnosti, koju je formulisao Albert Ajnštajn 1905. godine. Prema toj teoriji, zakoni fizike moraju biti isti u svim inercijalnim referentnim okvirima. Brzina svetlosti u vakuumu je konstantna za sve posmatrače, nezavisno od njihovog relativnog kretanja [6], [7].

U ovom okviru, ključni principi su očuvanje energije i očuvanje impulsa. Prema okviru, postoje i čestice koje imaju nultu masu, ali ipak poseduju energiju i impuls. Primer za takvu česticu je foton, kvant elektromagnetnog zračenja. On se uvek kreće brzinom svetlosti i uprkos tome što nema masu u mirovanju. Za ispitivanje malih objekata, uvodi se i kvantna fizika.

Kvantna fizika: Okvir kvantne fizike razjašnjava ponašanje atomskih i subatomskih čestica tako da svaki od tih objekata ima neko stanje (state) koji je definisan pomoću Šredingove talasne funkcije. Ove funkcije naznačuju neku energetska vrednost koja je reprezentovana pomoću talasa. A stanje čestica može da se menja pomoću rasejanja ili raspadanja i ove promene su tranzicije (transitions). Međutim, ove tranzicije nisu determinističke. Moguće je samo izračunati verovatnoću događaja tranzicije i tako novog stanja čestica [6], [7].

2.1. KVANTNA TEORIJA POLJA

Kombinovanjem temeljnih postulata specijalne teorije relativnosti i kvantne mehanike, razvija se kvantna teorija polja. Ovaj teorijski okvir omogućava opis ponašanja i međudejstva čestica u ekstremnim uslovima: pri vrlo visokim energijama i u izuzetno malim vremensko-prostornim razmerama. U tim uslovima, klasične i ne-relativističke teorije gube svoju primenjivost [6].

U kvantnoj teoriji polja, čestice se više ne posmatraju kao tačkasti entiteti sa sopstvenom egzistencijom, već kao pobuđeno stanje (ekscitacija) temeljnih kvantnih polja koja ispunjavaju prostor-vreme. Na primer, ono što nazivamo elektronom jeste kvantovana ekscitacija elektronskog polja, dok su fotoni, kvarkovi, gluoni i druge čestice slične ekscitacije odgovarajućih polja [6], [7], [8].

Relativistički deo ove teorije garantuje poštovanje prostor-vremenskih simetrija i zakona konzervacije koji iz njih proizlaze. To uključuje očuvanje energije i impulsa. Dozvoljeno je postojanje čestica bez mase, a nameće se ograničenje brzine prenosa informacije na brzinu svetlosti [6]. Kvantni aspekt teorije pruža slučajnu ili verovatnosnu prirodu fizičkih procesa. Tranzicije između stanja nisu determinističke, već su opisane statistički pomoću amplituda verovatnoće.

Teorija predviđa još i postojanje antičestica, kao što su pozitroni i antikvarkovi, koje su eksperimentalno potvrđene još sredinom 20. veka [6]. Još jedno važno svojstvo kvantne teorije polja jeste njena sposobnost da opiše sile kao posledicu razmene virtuelnih čestica:

elektromagnetna sila se tumači kao razmena fotona, jaka sila kao razmena gluona, a slaba kao razmena W i Z bozona [6], [7], [8].

2.2. STANDARDNI MODEL

Standardni model predstavlja vrhunac razvoja kvantne teorije polja. On je trenutno najuspešniji teorijski okvir za opisivanje mikroskopskih čestica i njihovih interakcija. U njegovoj osnovi se nalazi pretpostavka da su sve elementarne čestice u stvari pobuđeno stanje temeljnih kvantnih polja, a sile su posledica lokalnih simetrija, što znači da se transformacije mogu primenjivati nezavisno u svakoj tački prostor-vremena. Njihovo uvođenje implicira postojanje interakcionih polja, odnosno bozona koji posreduju u interakcijama [6].

Čestice koje učestvuju u interakcijama dele se na dve kategorije: fermione i bozone.

Fermioni: Fermioni su čestice sa spinom $1/2$ i oni predstavljaju osnovne gradivne jedinice materije. Fermioni se dalje klasifikuju u kvarkove i leptone, pri čemu oba tipa dolaze u po tri generacije. Kvarkovi – up, down, charm, strange, top i bottom – nikada ne postoje slobodno u prirodi, već se zbog jake sile uvek nalaze vezani u hadrone, odnosno protone, neutrone, mezone i druge kombinacije. S druge strane, leptoni – uključujući elektron, mion, tau i njihove odgovarajuće neutrine – mogu se uočiti kao slobodne čestice. Oni podležu slabim i elektromagnetnim silama (osim neutrina, on je neutralan i ima interackciju isključivo slabom silom) [7].

Bozoni: Bozoni su čestice sa celobrojnim spinom. Posreduju u interakcijama između fermiona. Foton, čestica bez mase koja prenosi elektromagnetnu silu, u potpunosti je opisan kvantnom elektrodinamikom. Gluoni, kojih ima osam, prenose jaku interakciju između kvarkova i imaju osobinu „boje“, a to je kvantni broj karakterističan za kvarkove u okviru kvantne hromodinamike [6]. Slaba sila se prenosi pomoću masivnih W^+ , W^- i Z^0 bozona, koji imaju značajnu ulogu u procesima radioaktivnog raspada i u transformaciji jednog kvarka u drugi, a to je osnova promena ukusa kvarkova u standardnim interakcijama [7].

Uvođenje Higsovog mehanizma unutar Standardnog modela bilo je neophodno kako bi se objasnila masa W i Z bozona. Bez tog mehanizma, svi bozoni bi morali biti bez mase, što nije u saglasnosti sa eksperimentima. Higsov mehanizam koristi spontano narušavanje simetrije, gde polje ne nultog vakuumskeg očekivanja generiše efektivne mase za čestice koje su s njim u interakciji [6], [1].

Pored toga što objašnjava interakcije između čestica i njihov spektar, Standardni model je izuzetno uspešan u predviđanju kvantitativnih rezultata koji se precizno potvrđuju u eksperimentima visoke energije. Na primer, merenja anomalnog magnetskog momenta elektrona slažu se sa teorijskim predviđanjima na nivou od 11 decimala, što je jedno od najtačnijih poređenja teorije i eksperimenta u istoriji fizike [6].

Uprkos svojoj preciznosti, Standardni model nije teorija svega. On ne uključuje gravitaciju, niti objašnjava poreklo tamne materije i tamne energije. Neutrini u okviru modela su bez mase, dok eksperimentalni rezultati potvrđuju da ipak imaju veoma malu, ali nenultu masu – što implicira da model mora biti proširen [7]. Takođe, model ne daje objašnjenje za asimetriju materije i antimaterije u svemiru, niti za vrednosti većine osnovnih konstanti koje unosi kao parametre.

Najveći test Standardnog modela bila je potraga za Higsovim bozonom. Njegovo postojanje bilo je ključno za potvrdu mehanizma masivnosti unutar teorije, a time i same konzistentnosti modela. Zato je Higsov bozon najvažnija čestica modela: ne zbog njegove mase ili interakcija, već jer osigurava unutrašnju koherenciju teorijskog okvira [1], [9].

2.3. HIGSOV BOZON

Higsov bozon je teorijska posledica mehanizma spontanog narušavanja simetrije unutar Standardnog modela, koji omogućava masivne čestice bez narušavanja lokalne simetrije (baždarske simetrije). Problem mase je, iz teorijske perspektive, bio jedan od najsloženijih u kvantnoj teoriji polja.

Osnovna ideja Higsovog bozona jeste postojanje skalarne kompleksne funkcije polja sa četiri realne komponente, od kojih tri „bivaju apsorbovane“ od strane W^+ , W^- i Z^0 bozona, čime oni stiču masu, dok četvrta komponenta ostaje kao fizičko kvantno pobuđeno stanje – Higsov bozon [6]. Ovaj proces se često naziva „spontano narušavanje simetrije“, pri čemu sistem bira specifičan vakuum među beskonačno mnogo simetričnih stanja, što dovodi do pojave efektivne mase čestica. Higsovo polje je u tom smislu univerzalno u prostoru-vremenu i predstavlja jedinstvenu pozadinu sa kojom sve druge čestice vrše interakcije. Jačina te interakcije određuje masu konkretne čestice: što je interakcija jača, masa je veća [7], [10], [11].

Iako je postojanje Higsovog bozona bilo teorijski utemeljeno još 1964. godine, eksperimentalna potvrda izostajala je gotovo pet decenija. Očekivalo se da će čestica imati izuzetno kratak životni vek, malu verovatnoću proizvodnje i složene kanale raspada, što je činilo njeno otkrivanje tehnički zahtevnim. Ali razvojem Velikog hadronskog sudarača (LHC) u CERN-u, i njegovih detektora CMS i ATLAS, stvoreni su uslovi za detekciju čestice sa osobinama koje odgovaraju predikcijama za Higsov bozon.

U julu 2012. godine, dve nezavisne kolaboracije objavile su da su identifikovale novu česticu mase približno 125 GeV koja se raspada u dva fotona ($H \rightarrow \gamma\gamma$), u dva Z bozona koji dalje prelaze u četiri leptona ($H \rightarrow ZZ^* \rightarrow 4\ell$), kao i u druge kanale poput $H \rightarrow WW^*$ i $H \rightarrow \tau^+\tau^-$ [1], [9]. Ovi raspadni omogućili su preciznu rekonstrukciju mase čestice i upoređivanje sa očekivanom pozadinom. Višak događaja u ovim kanalima bio je statistički značajan na nivou od pet standardnih devijacija, što u fizici čestica predstavlja prag za zvanično otkriće [1].

Dalje analize pokazale su da nova čestica ima spin nula, kao i pozitivan paritet, što je u saglasnosti sa skalarnom prirodom Higsovog bozona. Raspad u dva fotona eliminiše mogućnost spina 1, dok frekvencije raspada u različite kanale omogućavaju procenu jačine Higsove interakcije sa drugim česticama – uključujući kvarkove, leptone i bozone [1]. Na osnovu prikupljenih podataka zaključeno je da je reč o čestici čije su karakteristike saglasne sa predviđanjima Standardnog modela, iako precizna merenja još uvek traju.

Otkriće Higsovog bozona zaokružilo je teorijsku strukturu Standardnog modela, ali istovremeno je i otvorilo nova pitanja. Jedno od njih je pitanje jedinstvenosti Higsovog polja – da li postoji samo jedno skalarsko polje ili više njih? Takođe, potencijalna odstupanja u jačini interakcije Higsovog bozona sa ostalim česticama mogla bi ukazati na postojanje nove fizike izvan Standardnog modela. Istraživanja takođe ispituju da li se Higsovo polje može povezati sa inflacijom ranog svemira ili sa mehanizmom generisanja mase neutrina kroz takozvani seesaw mehanizam [7].

Eksperimentalna potvrda postojanja Higsovog bozona predstavlja jedan od najvažnijih naučnih uspeha XXI veka. Tim se potvrđuje ključna komponenta jedne od najpreciznijih teorija u fizici. Takođe, postavljaju se temelji za nova pitanja i eksperimente koji bi mogli proširiti naše razumevanje fundamentalnih zakona prirode.

2.4. BIG DATA U FIZICI ČESTICA

Zapremina, složenost i brzina nastajanja podataka u savremenoj eksperimentalnoj fizici je dostigla razmere koje se karakterišu kao Big Data. Termin „Big Data“ ne odnosi se samo na veliku količinu podataka, već i na niz drugih izazova koji prate njihovo generisanje, skladištenje, analizu i interpretaciju. Prema klasifikaciji iz industrije, osnovne karakteristike

Big Data sistema mogu se svesti na tzv. „5V”: volumen (volume), brzina (velocity), raznovrsnost (variety), verodostojnost (veracity) i vrednost (value) [12], [13].

U kontekstu fizike čestica, volumen podataka generisanih tokom jednog eksperimenta na LHC-u dostiže i do nekoliko petabajta godišnje. Samo CMS detektor proizvodi u proseku do 40 miliona sudara u sekundi. Od toga se selektuje i trajno skladišti samo mali deo (~1000 događaja/s), ali i to predstavlja ogroman izazov za softversku i hardversku infrastrukturu [13].

ROOT framework predstavlja jednu od najvažnijih karika u sistemu analize Big Data podataka u fizici. Zbog sposobnosti kompresije, strukturiranog čuvanja i selektivnog pristupa događajima, ROOT framework je savršen alat za obradu podataka koje generišu detektori visoke rezolucije, kao što je CMS. Međutim, sa porastom kompleksnosti i veličine podataka, standardne metode analize postaju sve više ograničene. Odatle se razvija potreba za integracijom naprednih Big Data tehnologija u analitički ekosistem fizike čestica [12].

2.4.1. ZNAČAJ BIG DATA TEHNOLOGIJA U FIZICI

Ključni izazovi u domenu fizike čestica proizlaze iz činjenice da eksperiment poput CMS-a proizvodi više desetina miliona čestičnih sudara u sekundi, ali je fizički i tehnički moguće sačuvati tek mali deo tih događaja. Potrebno je razviti algoritme koji će u realnom vremenu odabrati relevantne informacije, a kasnije omogućiti njihovu obradu i analizu u skalabilnom okruženju. Tako Big Data postaje neophodan za efikasnu filtraciju i dubinsku interpretaciju fizičkih signala [12], [13].

Na primer, analiza raspada Higsovog bozona u četiri leptona ($H \rightarrow ZZ^* \rightarrow 4\ell$) zahteva precizno rekonstruisanje tragova čestica unutar detektora. Ovaj zadatak uključuje obradu podataka iz više nezavisnih podsistema (elektromagnetski kalorimetar, mionski sistem, sistem za praćenje tragova), njihovo povezivanje u koherentne događaje, i primenu metoda statističkog testiranja za identifikaciju verovatnoće da posmatran signal potiče iz Higsovog bozona, a ne iz pozadine. Bez infrastrukture koja omogućava efikasno skladištenje i pristup ogromnim datasetovima, ovakve analize bi bile praktično neizvodljive [12].

Uz to, Big Data pristupi obezbeđuju ključne mogućnosti:

- Upravljanje masivnim, često distribuiranim datasetovima u realnom vremenu, čime se podržava analiza eksperimenata čija količina sirovih podataka prevazilazi petabajte godišnje.
- Redukcija i transformacija podataka pre nego što dođu u ruke fizičara – to uključuje formate optimizovane za brzu pretragu, filtere za uklanjanje nekonzistentnih događaja, a i automatizovano označavanje podataka putem prethodno obučениh modela.
- Paralelna obrada na heterogenim sistemima, uključujući GRID infrastrukturu i cloud okruženja, omogućava da se više datasetova simultano analizira ili ukrsti.
- Primena tehnike mašinskog učenja, kao što su klasifikacija pomoću neuronskih mreža, BDT (Boosted Decision Trees), ili detekcija anomalija, pomaže u identifikaciji retkih događaja i optimizaciji selekcije fizikalnih kanala.

Pored toga, Big Data pristupi značajno doprinose i u fazama koje prethode analizi, kao što su kontrola kvaliteta podataka, kalibracija detektora, praćenje vremenske stabilnosti i provera konzistentnosti između različitih eksperimentalnih kampanja. Sve više zadataka koji su se ranije obavljali ručno sada se automatizuju, čime se ubrzava ciklus naučnog otkrića.

Kao konkretan primer primene Big Data paradigme u fizici čestica, CERN je 2018–2019. godine započeo eksperimentalnu upotrebu Apache Spark platforme za obradu CMS podataka. Spark omogućava in-memory obradu, distribuciju zadataka i ubranu evaluaciju fizikalnih

promenljivih, što se pokazalo naročito korisnim u iterativnim analizama kao što je evaluacija efikasnosti selekcije događaja u Higsovima kanalima [12]. Uz to, razvoj Spark-ROOT konektora omogućio je direktnu interakciju između tradicionalnih .root fajlova i modernog Spark okruženja, čime je uklonjen jaz između specijalizovanog naučnog softvera i generičkih Big Data rešenja.

Razvojem budućih eksperimenata sa još većom frekvencijom događaja, kao što su planirane nadogradnje LHC-a, uloga Big Data tehnologija postaće još značajnija. Biće neophodno efikasnije obrađivanje i semantičko povezivanje podataka, kao i analiza pomoću veštačke inteligencije u realnom vremenu. Ti podaci biće integrisani sa podacima iz simulacija, teorijskih predikcija i metapodataka o eksperimentu.

2.4.2. TEHNOLOGIJE I UPOTREBA DANAS

Savremena analiza podataka u fizici čestica ne bi bila moguća bez oslanjanja na kompleksan softverski ekosistem koji spaja domene tradicionalne naučne obrade sa savremenim Big Data tehnologijama. U poslednjoj deceniji, CERN intenzivno radi na integraciji alata kao što su Apache Spark, Hadoop, Jupyter, Zeppelin i različiti mašinski algoritmi. Cilj je da se omogući skalabilna, distribuirana i efikasna analiza na petabajtskim datasetovima koje generišu eksperimenti poput CMS i ATLAS.

Jedan od najznačajnijih primera jeste integracija Apache Spark platforme u analitičke tokove CMS kolaboracije. Spark omogućava paralelnu obradu ogromnog broja ROOT fajlova raspoređenih na različitim čvorištima u mreži. Zahvaljujući in-memory arhitekturi, Spark izvršava operacije bez potrebe za stalnim upisivanjem i čitanjem podataka sa diska. Time se značajno ubrzavaju zadaci poput filtracije, selekcije i grupisanja događaja. U kontekstu fizike, to znači da se može vrlo brzo, kroz jednostavne SQL upite ili PySpark skripte, selektovati napr. sve sudare u kojima se u finalnom stanju pojavljuju četiri leptona sa određenim energetske pragom [12].

Ključni tehnički proboj bio je razvoj tzv. Spark-ROOT konektora, koji omogućava direktno učitavanje ROOT fajlova u Spark DataFrame objekte. To je značajan korak jer omogućava da se tradicionalni format iz fizike čestica (koji koristi hijerarhijske strukture kao što su TTrees i n-tuples) transformiše u strukture koje su pogodnije za obradu u Spark i Pandas okruženju [12]. Ovo je posebno korisno za mlade istraživače koji nisu nužno stručnjaci za C++ ili ROOT, ali imaju veštine u radu sa Python, SQL (Structured Query Language) ili mašinskim učenjem.

Osim Spark-a, CERN sve više koristi Hadoop ekosistem za trajno skladištenje i indeksiranje podataka. Hadoop Distributed File System (HDFS) koristi se za čuvanje datasetova koji mogu da dostignu desetine petabajta, a u kombinaciji sa Hive, Presto ili Impala sistemima moguće je izvršiti veoma brze pretrage i agregacije nad tim podacima.

Za interaktivnu vizualizaciju, alati kao što su Jupyter Notebook i Apache Zeppelin omogućavaju korisnicima da kreiraju skripte koje je moguće reprodukovati. Oni kombinuju kod, rezultate i objašnjenja u istom dokumentu. Takvi dokumenti se sve više koriste za dokumentaciju analiza, evaluaciju metoda i naučnu kolaboraciju unutar velikih timova. Vizualizacije kao što su histogrami mase čestica, scatter plotovi međuzavisnosti ili prikazi vremenskog ponašanja detektora mogu se kreirati u realnom vremenu, čak i nad velikim podacima.

Poseban segment koji doživljava ubrzan razvoj jeste primena mašinskog učenja (ML) u obradi eksperimentalnih podataka. Klasične metode analize u fizici čestica zamenjuju se sofisticiranim klasifikacionim i regresionim modelima. Boosted Decision Trees (BDT), Random Forests i deep learning modeli koriste se za detekciju retkih događaja (kao što su egzotični raspad ili supersimetrične čestice), selekciju signala u visokoj pozadini, kao i za automatsku rekonstrukciju parametara čestica na osnovu sirovih detektorskih podataka.

Na primer, integracija Spark MLlib biblioteke sa CMS datasetovima, pri čemu su se modeli obučavali za klasifikaciju između Higsovih događaja i standardne pozadine. Analize ovog tipa

postižu veću tačnost u razdvajanju signala od šuma. Takođe, one omogućavaju transparentno merenje verovatnoće pogrešne klasifikacije, a to je od suštinskog značaja u visoko preciznim merenjima [12].

U projektima koji istražuju nepoznatu fiziku, poput dark matter pretraga ili novih interakcija, koristi se detekcija anomalija uz pomoć nekontrolisanog učenja. Treniraju se na poznatoj pozadini i zatim skeniraju datasetove u potrazi za neuobičajenim događajima koji bi mogli ukazati na novu fiziku.

Pored direktne upotrebe u analizama, ML algoritmi nalaze ratuću primenu u podršci eksperimentima: u real-time selekciji događaja (trigger sistemi), automatskoj detekciji grešaka u detektorima, kalibraciji parametara i predikciji performansi. Big Data u kombinaciji sa ML postaje ključna osovina budućeg razvoja kako eksperimentalne infrastrukture, tako i teorijskog istraživanja.

3. ROOT FRAMEWORK

Veliki hadronski sudarač (LHC), kao najmoćniji akcelerator čestica na svetu, generiše ogromne količine eksperimentalnih podataka. Tokom prethodnih decenija, CERN je razvio više softverskih rešenja za obradu podataka, kao što su PAW (Physics Analysis Workstation), ali nijedno nije moglo da ispuni sve zahteve koje je postavio LHC u pogledu obima, brzine obrade i fleksibilnosti. Do sredine 1990-ih godina bilo je jasno da je potreban novi, robustan softverski okvir sposoban za efikasnu obradu, analizu, skladištenje i vizualizaciju petabajta podataka koji potiču iz visokoenergetskih fizičkih eksperimenata.

Dvojica inženjera u CERN-u, Rene Brun i Fons Rademakers, počeli su 1995. godine da razvijaju novi softverski okvir, ROOT Framework. Prva javna verzija objavljena je 1997. godine kao open source projekat, što je omogućilo široku upotrebu i doprinos globalne zajednice istraživača i programera. Od tada, ROOT je postao standardni alat za analizu podataka u eksperimentima visokoenergetske fizike, posebno u okviru CERN-ovih kolaboracija poput CMS i ATLAS [14].

ROOT Framework je dizajniran kao objektno-orijentisan sistem zasnovan na programskom jeziku C++, što omogućava direktan rad sa kompleksnim strukturama podataka i visoku efikasnost. Osnovu ROOT-a čine stotine međusobno povezanih klasa koje korisnicima omogućavaju:

- rad sa velikim skupovima podataka,
- izvođenje naprednih statističkih analiza,
- kreiranje sofisticiranih vizualizacija,
- upravljanje datotekama i bazama podataka specifičnim za fiziku čestica.

Kako bi se olakšao rad i ubrzao razvoj, ROOT uključuje C++ interpreter (Cling) koji omogućava interaktivno testiranje koda, ali i kompajliranje za produkcionu rad. Ova fleksibilnost čini ROOT pogodnim za sve faze analize, od eksperimentisanja sa idejama do izrade finalnih rezultata spremnih za objavljivanje.

Jedna od ključnih prednosti ROOT-a je integrisana podrška za paralelnu obradu podataka, što značajno ubrzava simulacije i analize. To omogućava istraživačima da razvijaju modele ponašanja čestica i analiziraju signale koji ukazuju na retke fizičke pojave, bez potrebe za neprekidnim pokretanjem složenih eksperimenata u LHC-u [15].

Još jedna fundamentalna komponenta ROOT-a je njegova sopstvena hijerarhijska baza podataka, zasnovana na tzv. ROOT fajlovima (.root). Ovi fajlovi omogućavaju efikasno čuvanje i prenos složenih C++ objekata uz automatsku kompresiju podataka, što rezultira značajnim smanjenjem veličine fajlova, bez gubitka informacija. ROOT fajlovi mogu sadržavati ne samo podatke, već i metapodatke, dijagrame i konfiguracije analize, čineći ih idealnim za deljenje rezultata i replikaciju analiza.

Osim osnovne funkcionalnosti, ROOT podržava i naprednu vizualizaciju pomoću dvodimenzionalnih i trodimenzionalnih grafikona, koji se mogu izvesti u raster ili vektorske formate. Takođe uključuje biblioteke za statistiku, linearne algebre, numeričku obradu i mašinsko učenje, što ga čini izuzetno sveobuhvatnim okvirom [15].

ROOT je dodatno proširiv zahvaljujući podršci za druge programske jezike, naročito Python (putem PyROOT-a), što omogućava lakšu integraciju u savremene tokove razvoja i analize. Postoji i podrška za jezik R, koji se koristi za statističku obradu, kao i različiti alati za povezivanje sa sistemima poput Apache Spark-a u okviru Big Data tehnologija, što je opisano u literaturi [12], [16].

Zahvaljujući svom modularnom i proširivom dizajnu, ROOT Framework se koristi ne samo u fizici čestica, već i u drugim oblastima koje zahtevaju obradu velikih količina naučnih podataka, npr. u astrofizici i bioinformatiči.

3.1. ARHITEKTURA ROOT OKVIRA

ROOT Framework je razvijen kao modularni objektno-orijentisani softverski sistem, zasnovan na programskom jeziku C++. Njegova arhitektura omogućava visok stepen fleksibilnosti, proširivosti i efikasnosti u radu sa velikim naučnim podacima, što je naročito značajno u oblasti visokoenergetske fizike. Sistem je organizovan tako da korisnik ima pristup različitim funkcionalnim komponentama putem hijerarhijski strukturisanih biblioteka, pri čemu se učitavaju samo one biblioteke koje su relevantne za konkretnu analizu ili primenu [14], [15].

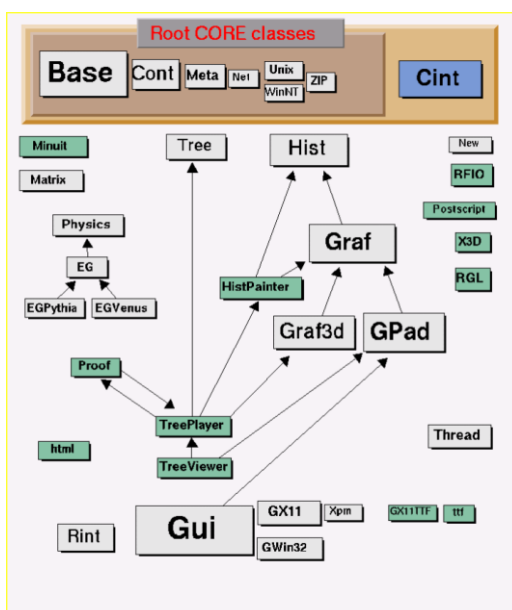
Centralni koncept arhitekture ROOT-a jeste raspodela funkcionalnosti kroz specijalizovane biblioteke, pri čemu svaka biblioteka sadrži skup povezanih klasa koje se odnose na određeni domen rada. Ovakva modularizacija omogućava da aplikacije ostanu kompaktne i efikasne jer korisnik može eksplicitno da koristi samo one komponente koje su mu potrebne, dok ostatak sistema ostaje neaktivan.

Dve osnovne i obavezne biblioteke u svakoj ROOT aplikaciji su:

libCore: Ovo je osnovna biblioteka ROOT Framework-a i predstavlja temeljnu komponentu svakog ROOT programa. Sadrži klase koje omogućavaju osnovne operacije kao što su upravljanje objektima, kontejneri (itd TList, TObjArray), rukovanje metapodacima (pomoću TClass), upravljanje fajlovima, sistemska nezavisnost i osnovni algoritmi za kompresiju podataka (npr. implementacija ZIP algoritma). Bez ove biblioteke rad u ROOT-u nije moguć, jer ona čini jezgro svih aplikacija.

libCling (ranije *libCint*): Ova biblioteka predstavlja interpreter C++ jezika integrisan u ROOT Framework. Omogućava korisnicima da interaktivno izvršavaju C++ kod bez potrebe za kompajliranjem. Ovo je naročito korisno u fazama razvoja, testiranja i vizuelizacije, jer omogućava brzu iteraciju nad analizom bez pokretanja kompletnog build procesa. Interpreter se bazira na LLVM/Clang tehnologiji i omogućava kombinaciju interpretiranog i kompajliranog koda unutar istog okruženja [15].

Interpreter i osnovne klase zajedno čine tzv. ROOT Core sloj, koji je vizualno označen crvenom bojom u zvaničnoj arhitektonskoj šemi (videti sliku 1). Svi ostali slojevi i biblioteke sistema oslanjaju se na ovaj temelj.



Slika 1. Arhitektura ROOT okvira (izvor: [14])

ROOT koristi hijerarhijski model zavisnosti između biblioteka. Kada korisnik eksplicitno uključi neku višu biblioteku (npr. libHist za histograme), automatski se učitavaju i sve biblioteke koje se nalaze niže u hijerarhiji i od kojih ta biblioteka zavisi. Strelice u arhitektonskoj dijagrami ukazuju na smer zavisnosti – od viših ka nižim bibliotekama [15].

Pored eksplicitnog učitavanja, mnoge biblioteke se učitavaju dinamički, na zahtev – tj., kada se u korisničkom kodu pozove neka specifična klasa ili funkcija, sistem automatski detektuje potrebnu biblioteku i učitava je. Ovo omogućava dodatnu optimizaciju u pogledu memorijskog otiska i brzine pokretanja aplikacija.

Na slici 1 prikazana je arhitektura ROOT Framework-a kroz dijagram zavisnosti između njegovih ključnih komponenti, odnosno biblioteka i modula. Biblioteke koje se učitavaju automatski označene su zelenom bojom, dok su ostale prikazane sivim nijansama. Dinamičko učitavanje i hijerarhijska zavisnost olakšavaju modularni razvoj i upotrebu ROOT-a u distribuiranim sistemima.

Na vrhu dijagrama nalaze se ROOT Core klase, koje obuhvataju osnovne funkcionalnosti sistema:

- Base – temeljne klase kao što su TObject, TNamed, i druge osnovne C++ klase,
- Cont - kolekcije i kontejneri (TList, TMap, TClonesArray),
- Meta – klase za upravljanje meta-informacijama (TClass, TMethod),
- Net – mrežna komunikacija i podrška za distribuciju,
- Unix/WinNT – sloj nezavistan od operativnog sistema,
- ZIP – kompresioni algoritmi za ROOT fajlove.

Paralelno sa baznim klasama, sa desne strane je prikazana Cint/Cling biblioteka (plavo obojena), koja omogućava interaktivnu interpretaciju C++ koda. Ova komponenta je odvojena ali čvrsto povezana sa osnovnim slojem i omogućava dinamičko izvršavanje bez prethodnog kompajliranja [14], [15].

Iz osnovnih slojeva proizlaze složenije biblioteke:

- Tree – strukturirano skladištenje i pristup velikim skupovima podataka,
- Hist – podrška za jednorazmerne i višerazmerne histograme,
- Graf, Graf3D, GPad – grafička reprezentacija i crtanje funkcija, grafikona i interaktivnih panela,
- Gui – komponente korisničkog interfejsa za izradu vizuelnih alata i pregled rezultata analize.

Komponente koje se oslanjaju na ove biblioteke, prikazane niže u dijagramu, uključuju napredne funkcije poput:

- Rint – interaktivno okruženje ROOT-a,
- Proof – paralelna obrada podataka u distribuiranim sistemima,
- TreePlayer/TreeViewer – alati za rad sa TTree objektima,
- Html, Postscript, X3D, RGL – alati za eksport i vizualizaciju u različitim formatima,

- GX11, GWin32 – slojevi za grafički izlaz specifični za operativne sisteme.

Sve ove biblioteke proširuju osnovne funkcionalnosti ROOT-a i čine ga pogodnim za različite oblike analize, vizualizacije, kao i za izradu složenih algoritama.

3.2. ROOT FAJLOVI

Jedna od ključnih komponenti ROOT Framework-a jeste sopstveni sistem za skladištenje podataka zasnovan na ROOT fajlovima (.root). Ovi fajlovi predstavljaju komprimovane binarne strukture specijalno razvijene za efikasno čuvanje i razmenu velikih količina naučnih podataka, uz zadržavanje svih metapodataka potrebnih za potpunu rekonstruibilnost objekata unutar ROOT okruženja [14], [15].

ROOT fajlovi omogućavaju čuvanje različitih tipova sadržaja, uključujući:

- numeričke i tekstualne podatke,
- C++ objekte bilo koje složenosti,
- hijerarhijske strukture podataka (poput stabala – trees),
- grafove, histograme i druge vizualne prikaze,
- informacije o konfiguraciji operativnog sistema i ROOT okruženju.

Budući da se radi o komprimovanim binarnim fajlovima, ROOT fajlovi zauzimaju značajno manje prostora u odnosu na nekompresovane tekstualne reprezentacije, što ih čini pogodnim za dugotrajno čuvanje i prenos putem mreže. Za implementaciju kompresije koristi se ZIP algoritam, koji je sastavni deo libCore biblioteke.

Zahvaljujući integraciji tzv. "rečnika" (dictionaries) – metapodataka koji opisuju strukturu i tipove čuvanih objekata – ROOT fajlovi omogućavaju automatsku rekonstrukciju objekata prilikom učitavanja. Rečnici su generisani automatski putem ROOT-ovog sistema za refleksiju (TClass, TStreamerInfo) i omogućavaju da se binarni podaci mapiraju nazad u C++ objekte bez potrebe za ručnim definisanjem strukture podataka. Ovo je od ključnog značaja u složenim analizama visokoenergetskih eksperimenata, gde se često radi sa veoma kompleksnim događajima [15].

Važno je napomenuti da je sintaksa za čitanje i pisanje ROOT fajlova otvorena i dokumentovana, što omogućava pristup i drugim aplikacijama van ROOT okruženja. Ovo olakšava integraciju sa drugim softverskim alatima i analitičkim platformama.

ROOT fajlovi podržavaju internu hijerarhiju, slično fajl sistemima operativnih sistema. Unutar jednog ROOT fajla mogu se nalaziti poddirektorijumi, pa čak i drugi ROOT fajlovi, čime se omogućava organizacija velikih i složenih podataka na logički strukturisan način. To doprinosi boljoj modularnosti i skalabilnosti analiza.

Podaci unutar ROOT fajlova mogu biti organizovani na različite načine, pri čemu se najčešće koristi struktura poznata iz baza podataka – n-torke (n-tuples), ili naprednija struktura – drveće (trees).

Kada se koristi pristup n-torkama, podaci su organizovani u formi tabela:

- redovi predstavljaju pojedinačne događaje (events),
- kolone predstavljaju promenljive (variables).

Iako intuitivna i jednostavna, ova organizacija ima ograničenja u efikasnosti, posebno kod velikih datasetova. Naime, ukoliko korisniku nije potreban ceo događaj, već samo nekoliko

savremenim istraživanjima u oblasti fizike visokih energija, ali i u drugim naučnim disciplinama koje se oslanjaju na obradu velikih datasetova [14], [15].

ROOT sadrži implementaciju svih standardnih matematičkih funkcija iz C i C++ jezika, ali dodatno proširuje svoje mogućnosti putem specijalizovanih biblioteka kao što su MathCore, MathMore i GSL (GNU Scientific Library). Ove biblioteke omogućavaju:

- rad sa vektorima i matricama, uključujući operacije linearne algebre,
- izvođenje numeričke integracije i diferencijacije,
- rešavanje sistemâ jednačina i nelinearnih funkcija,
- nalaženje minimuma i maksimuma višedimenzionalnih funkcija,
- interpolaciju i aproksimaciju podataka.

ROOT takođe omogućava rad sa geometrijskim entitetima – npr. tačkama, pravcima i vektorima – uključujući 4-vektore (TLorentzVector) koji se koriste u relativističkoj fizici za predstavljanje energije i impulsa čestica. Omogućena je i transformacija koordinatnih sistema između 2D, 3D i 4D prostora, što je ključno za simulacije u visokoenergetskoj fizici i detekciji čestica [15].

Jedna od najvažnijih funkcionalnosti ROOT-a jeste sposobnost da se vrši simulacija fizičkih događaja. ROOT sadrži biblioteke za generisanje slučajnih i pseudo-slučajnih brojeva, uz implementaciju tri napredna algoritma:

- RANLUX (visokokvalitetni generator sa unapređenom uniformnošću),
- L'Ecuyer kombinovani generator (pogodan za simulacije u više dimenzija),
- MT19937 (Mersenne Twister) – poznat po velikom periodu i statističkoj preciznosti [15].

Korisnik može generisati podatke koji prate određene statističke distribucije, uključujući najčešće korišćene:

- Gaussovu (normalnu) distribuciju,
- eksponencijalnu distribuciju,
- Poasonovu distribuciju,
- Landau distribuciju (često korišćenu u fizici čestica za opis gubitka energije).

Dodatno, ROOT podržava generisanje uzoraka na osnovu empirijskih distribucija sa histograma, koristeći biblioteku UNU.RAN, što omogućava simulaciju podataka u skladu sa prethodno uočenim statističkim obrascima [15].

Zbog obima podataka koji se analiziraju u visokoenergetskoj fizici, ROOT sadrži ugrađenu podršku za paralelnu obradu podataka putem alata PROOF (Parallel ROOT Facility). Ovaj alat omogućava raspodelu analize na više čvorova u klasteru, čime se:

- smanjuje vreme obrade velikih datasetova,
- optimizuje korišćenje dostupnih računarskih resursa,
- omogućava skalabilna analiza na grid i cloud infrastrukturnama [13], [16].

Podaci se automatski dele između čvorova, svaki čvor vrši obradu svog dela dataset-a, a rezultati se zatim kombinuju u jedinstven izlaz.

ROOT omogućava analizu podataka pomoću C++ makroa, koji se mogu definisati, testirati i izvršavati u interpretativnom režimu pomoću Cling interpreter-a. Ovo omogućava brzo testiranje hipoteza i modifikaciju analiza bez potrebe za eksplicitnim kompajliranjem.

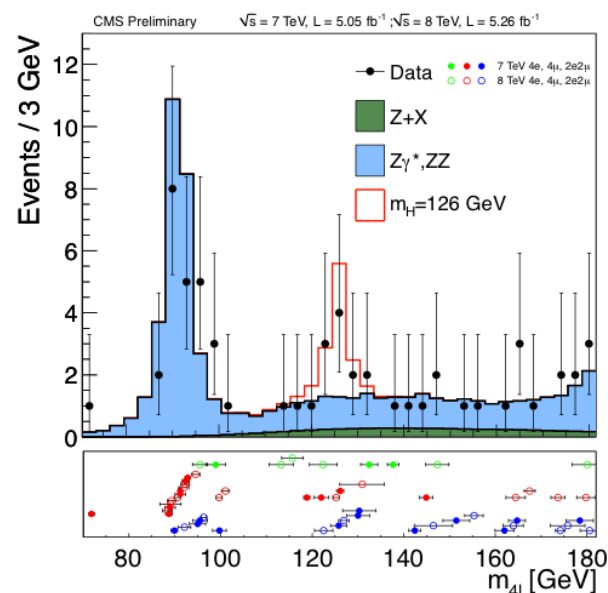
Centralni alat za vizuelnu analizu podataka u ROOT-u su grafikoni i histogrami, koji se mogu:

- interaktivno prikazivati,
- eksportovati u različite formate (npr. .jpg, .png, .eps, .pdf),
- čuvati u ROOT fajlovima zajedno sa pripadajućim kodom, čime se omogućava kasnija replikacija i reprodukcija grafika.

Pored histograma, ROOT nudi podršku za

- 1D, 2D i 3D grafove (TGraph, TGraph2D, TH3),
- konturene dijagrame,
- funkcijske prikaze (analitičke i numeričke funkcije),
- višestruke slojeve prikaza (TPad, TCanvas) za kompleksne prezentacije.

Sve ove grafičke mogućnosti omogućavaju visoko prilagodljivu i informativnu prezentaciju rezultata, koja je od ključne važnosti pri objavljivanju naučnih rezultata.

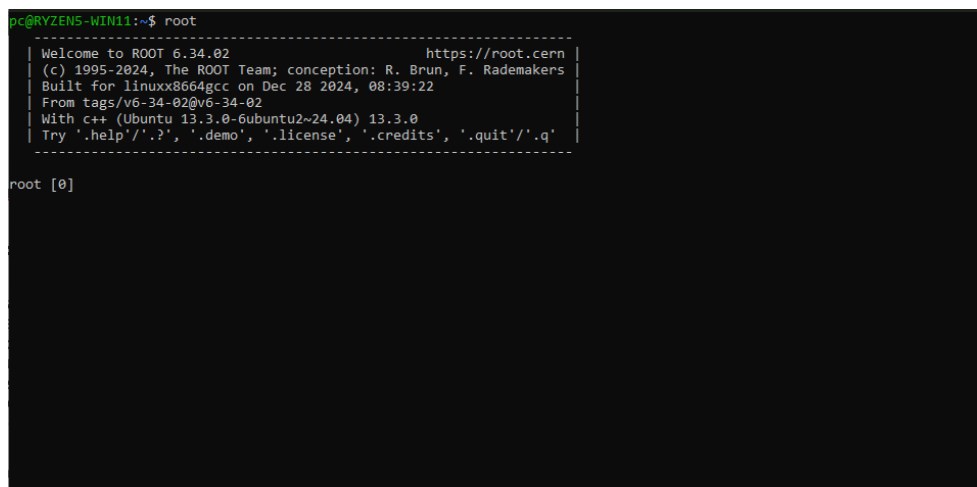


Grafik 1. Primer histograma u ROOT okviru (izvor: <https://root.cern/gallery/#data-analysis-and-visualization>)

3.4. UPOTREBA ROOT FRAMEWORK-A

ROOT Framework je dizajniran tako da bude višestruko prenosiv i može se instalirati na sve glavne operativne sisteme, uključujući Linux, Windows i macOS. Najpouzdaniji način instalacije je korišćenjem paket menadžera (npr. apt, brew, conda), ali su dostupni i unapred kompajlirani binarni fajlovi, kao i mogućnost ručne kompilacije iz izvornog koda, što omogućava maksimalnu kontrolu nad konfiguracijom sistema [15].

Na slici 3 prikazano je pokretanje ROOT Framework-a putem Windows Subsystem for Linux (WSL), u kojem je instalirana distribucija Ubuntu Linux. Pokretanjem komande `root` iz komandne linije otvara se ROOT interaktivna konzola (CLI), koja omogućava izvršavanje komandi i C++ izraza pomoću ugrađenog interpreter-a Cling [14].



```
pc@RYZEN5-WIN11:~$ root
-----
Welcome to ROOT 6.34.02                                     https://root.cern
(c) 1995-2024, The ROOT Team; conception: R. Brun, F. Rademakers
Built for linuxx86_64gcc on Dec 28 2024, 08:39:22
From tags/v6-34-02@v6-34-02
With c++ (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
Try '.help'/'?', '.demo', '.license', '.credits', '.quit'/'q'
-----
root [0]
```

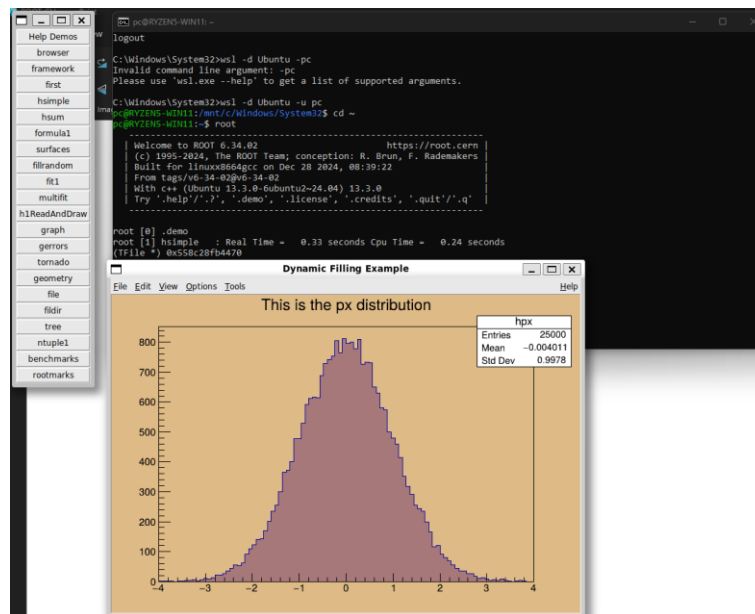
Slika 3. ROOT CLI

U početnom prozoru prikazane su osnovne informacije o instaliranoj verziji, uključujući verziju ROOT-a (npr. 6.34), platformu, datum kompilacije i kompajler. Unutar CLI-a dostupne su korisne komande:

- `.help` – prikaz pomoći,
- `.demo` – pokretanje demonstracionih primera,
- `.license` – prikaz licence,
- `.credits` – prikaz zaslužnih autora i saradnika,
- `.q` – izlazak iz sesije.

Zahvaljujući Cling-u, moguće je direktno uneti C++ kod, uključujući višelinijske izraze, koji se pišu unutar vitičastih zagrada, kao što je prikazano u komandnoj sesiji.

Na slici 4 prikazan je rezultat pokretanja `.demo` komande, koja otvara interaktivni demonstracioni prozor sa različitim primerima vizualizacije podataka. Jedan od primera učitava histogram iz datoteke `hsimple.root`, pri čemu se prikazuje distribucija promenljive `px` u obliku kolumnarnog histograma.

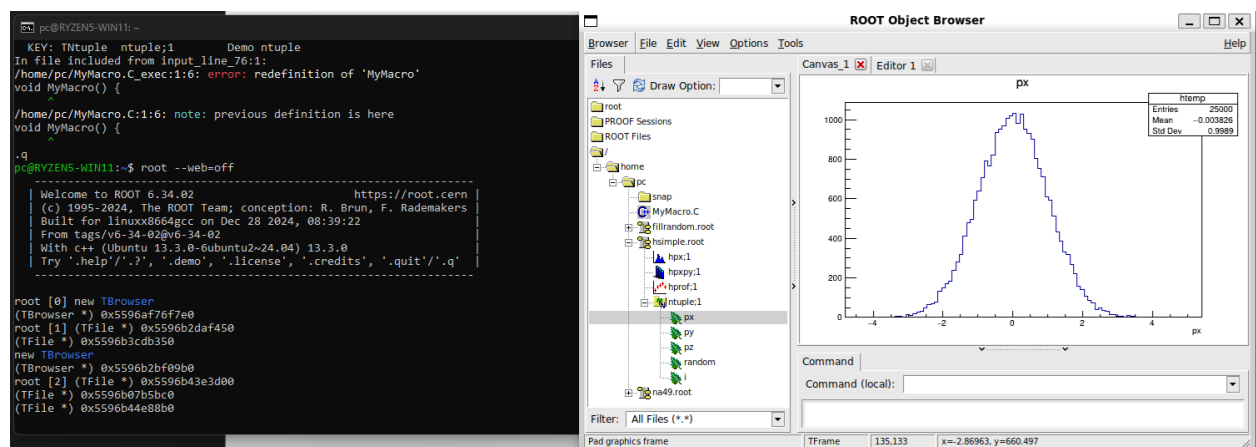


Slika 4. Demo prozor u ROOT-u

Histogram je generisan i vizualizovan automatski pomoću ROOT-ovog grafičkog sistema (TH1F i TCanvas klase), a sve funkcije su interaktivno dostupne – uključujući prikaz statistike i osnovnih parametara raspodele. Ovaj primer omogućava korisnicima brzo razumevanje načina na koji su podaci organizovani i prikazani.

Na slici 5 prikazano je pokretanje ROOT-a uz dodatak opcije `--web=off`, koja omogućava korišćenje standardnog GUI interfejsa unutar WSL okruženja. Nakon pokretanja komandne linije i unosa komande `new TBrowser()`, otvara se ROOT Object Browser, alat koji omogućava:

- vizualni pregled sadržaja ROOT fajlova,
- prikaz organizacije direktorijuma i podataka,
- kreiranje i manipulaciju histogramima,
- pregled i uređivanje C++ fajlova (makroa).



Slika 5. ROOT GUI

U prikazanom primeru, datoteka `hsimple.root` učitana je u pregled, i jasno je vidljiva njena struktura – sadrži jednu n-torku (`ntuple`) sa više promenljivih: `px`, `py`, `pz`, `random`. Svaka od ovih promenljivih može se pojedinačno vizualizovati klikom, što otvara odgovarajući

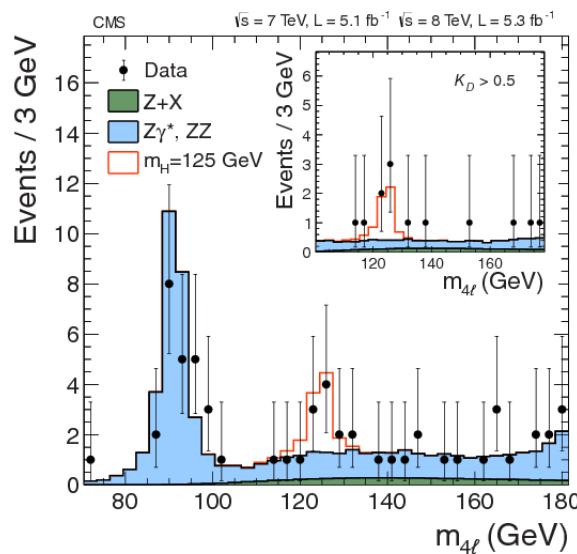
histogram. Ova struktura direktno prikazuje kako su podaci organizovani kolonijalno, što omogućava selektivno učitavanje i visoku efikasnost pri analizi.

Takođe, korisnik ima pristup Editor tabu u kojem može da piše ili uređuje C++ makroe (.C fajlovi), definiše sopstvene klase, funkcije ili celokupne biblioteke. Ovi makroi mogu da se izvrše direktno u GUI-ju, čime se omogućava brza iteracija kroz analitičke procese.

4. REPLIKACIJA ANALIZE PODATAKA

Jedan od najpreciznijih i najčistijih kanala za detekciju Higsovog bozona u okviru CMS eksperimenta jeste njegov raspad u četiri leptona putem dva Z bozona – jednog stvarnog i jednog virtuelnog. Ovaj proces se označava kao $H \rightarrow ZZ^* \rightarrow 4\ell$, gde ℓ predstavlja elektron ili mion. Ovaj kanal je izuzetno važan jer omogućava visoku masenu rezoluciju, malu pozadinu i potpunu rekonstrukciju finalnog stanja. Upravo zato je bio jedan od ključnih faktora u potvrđivanju postojanja Higsovog bozona u CMS eksperimentu 2012. godine [1].

Na dijagramu raspodele mase četiri leptona (grafik 2), eksperimentalni podaci su predstavljeni crnim tačkama koje označavaju učestanost detektovanih događaja za svaku vrednost mase rekonstruisanog sistema od četiri leptona. Na horizontalnoj osi prikazana je invarijantna masa sistema, dok je vertikalna osa broj događaja po energetsom opsegu.



Grafik 2. Objavljena vizualizacija rezultata (izvor: [1])

Plavo senčeno područje prikazuje pozadinu, odnosno distribuciju događaja za koje se očekuje da potiču iz poznatih procesa koji ne uključuju Higsov bozon – pre svega iz standardnog raspada para Z bozona nastalih iz kvark-antikvark sudara ($q\bar{q} \rightarrow ZZ \rightarrow 4\ell$). Ova distribucija dostiže maksimum kod mase od oko 91 GeV, što odgovara tačnoj masi jednog Z bozona. Ovaj vrh je fizički očekivan jer mnogi događaji uključuju barem jedan stvarni Z bozon u svom raspadu.

Međutim, ono što je ključno za potvrdu postojanja Higsovog bozona je pojava drugog, manjeg vrha kod mase od oko 125 GeV, koji se na dijagramu označava crvenom bojom. Ovaj dodatni maksimum ne može se objasniti standardnim pozadinskim procesima i precizno se uklapa u teorijska predviđanja za Higsov bozon. Prisustvo ovog signala ukazuje na to da postoji čestica koja se raspada u dva Z bozona, a zatim u četiri leptona, i čija masa odgovara 125 GeV – upravo onoj koju predviđa Standardni model za Higsov bozon [1].

4.1. NAČIN REPLIKACIJE ANALIZE

Da bi se replikovala analiza koja je dovela do ovog rezultata, CERN je omogućio javni pristup odgovarajućim podacima i softverskim alatima putem CERN Open Data portala. Detaljna uputstva za izvođenje ove replikacije nalaze se na sledećem linku: <https://opendata.web.cern.ch/record/5500>. U okviru ovog projekta koriste se Monte Karlo simulacije umesto stvarnih eksperimentalnih podataka, što je standardna praksa u visokoenergetskoj fizici kada se trenira analiza. Monte Karlo metod omogućava generisanje slučajnih događaja u skladu sa poznatim probabilističkim zakonima fizike čestica i realističnim modelima detektora.

Podaci su organizovani u datasetove koji odgovaraju različitim finalnim stanjima čestica. Za raspad $H \rightarrow ZZ^* \rightarrow 4\ell$, koristi se sledeća strategija:

- Događaji sa četiri miona (4μ) i kombinacijom dva elektrona i dva miona ($2e2\mu$) uzimaju se iz dataset-a DoubleMuParked.
- Događaji sa četiri elektrona ($4e$) uzimaju se iz dataset-a DoubleElectron.

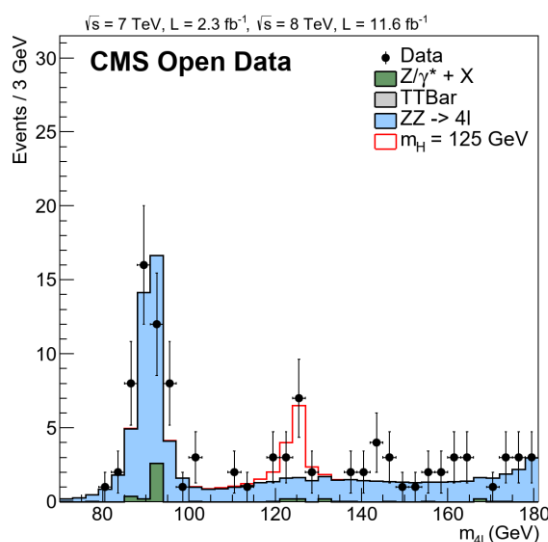
U sledećem koraku vrši se normalizacija podataka, kako bi se pozadina i signal mogli direktno uporediti. Pozadinski doprinos iz produkcije para Z bozona (ZZ) skalira se na maseni opseg od 180–600 GeV, dok se Higsova kontribucija skalira u signalnom opsegu od 80–180 GeV. Ovo omogućava da se Higsov signal statistički izdvoji iz pozadine, i da se precizno proceni njegova masa i učestalost pojavljivanja.

Kod koji se koristi za ovu replikaciju nije identičan originalnom kodu koji je koristila CMS kolaboracija tokom eksperimenta, već je pojednostavljen i optimizovan za edukativne svrhe. Ipak, dovoljno verno prikazuje ključne korake u analizi: učitavanje podataka, selekciju događaja, rekonstrukciju mase, vizualizaciju histograma i evaluaciju statističke značajnosti.

Kroz ovu replikaciju, korisnici imaju priliku da direktno eksperimentišu sa analitičkim postupcima koji se koriste u stvarnim naučnim istraživanjima u oblasti fizike čestica, da se upoznaju sa formatom i strukturom ROOT fajlova, i da razviju dublje razumevanje eksperimentalne potvrde Higsovog bozona.

4.2. PRVI NIVO REPLIKACIJE

Replikacija ima četiri nivoa. Prvi nivo obuhvata kvalitativno poređenje dijagrama iz grafika 2 i grafika 3, pri čemu se posmatraju sličnosti i razlike u obliku, skali i interpretaciji podataka. Na oba dijagrama prikazana je raspodela invarijantne mase četiri leptonona (4ℓ), koja predstavlja ključni kanal za detekciju raspada Higsovog bozona preko procesa $H \rightarrow ZZ \rightarrow 4\ell$. U oba slučaja centralna masa sudara iznosi 7 i 8 TeV, što odgovara uslovima rada Velikog hadronskog sudarača (LHC) u periodu 2011–2012. godine. Međutim, ukupna integrisana luminoznost se razlikuje: u originalnoj CMS publikaciji (grafik 2) koristi se puna količina prikupljenih podataka ($\sim 10 \text{ fb}^{-1}$), dok se u Open Data verziji (grafik 3) koristi manji i uprošćen skup podataka sa delimičnom simulacijom signala. Luminoznost predstavlja meru ukupnog broja sudara i direktno utiče na broj detektovanih događaja, što se ogleda u manjem statističkom uzorku i manjoj jasnoći signala u grafici 3.



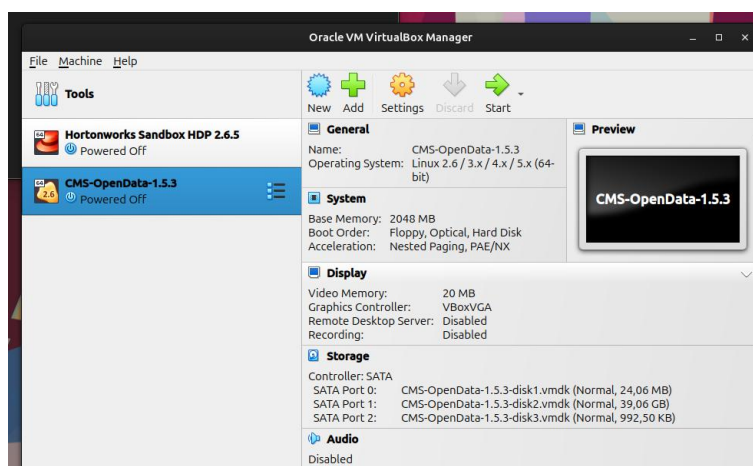
Grafik 3. Rezultat sa podacima iz Monte Karlo simulacije

Takođe, pozadinski doprinosi u zvaničnom CMS dijagramu (grafik 2) precizno su modelovani i prate eksperimentalne nesigurnosti, dok su u Open Data dijagramu (grafik 3) znatno uprošćeni. Uprkos tome, i u jednostavnijem dijagramu se uočava karakteristični „bump” oko 125 GeV, koji ukazuje na pojavu Higsovog bozona. Ovo pokazuje da i uz ograničene podatke i pojednostavljene metode moguće je dobiti približan oblik rezultata originalne analize. Time se ispunjava osnovni cilj prvog nivoa replikacije — vizuelna i konceptualna potvrda postojanja signala u kanalu četiri leptona pomoću javno dostupnih podataka.

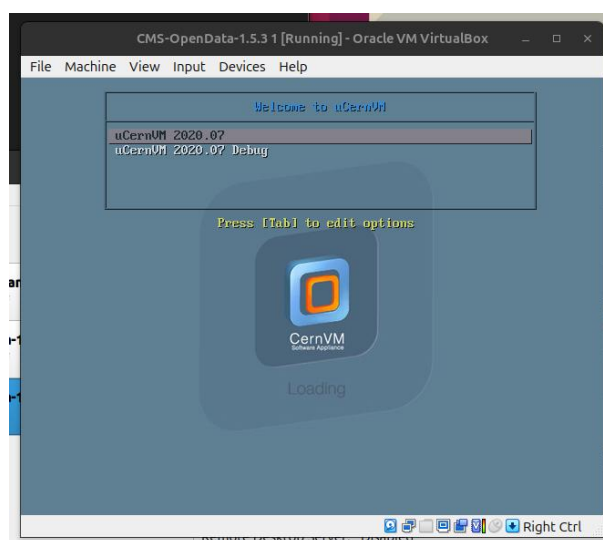
4.3. DRUGI NIVO REPLIKACIJE

Drugi nivo replikacije obuhvata reprodukciju referentnog dijagrama mase četiri leptona iz CMS analize raspada Higsovog bozona, poznatog kao `mass4l_combine`. Cilj ovog nivoa je da se, korišćenjem javno dostupnih ROOT fajlova i C++ makro koda, generiše dijagram koji precizno rekonstruiše histogram objavljen u originalnoj CMS publikaciji iz 2012. godine. Time se omogućava nezavisna provera validnosti i transparentnosti eksperimentalnih podataka.

Za potrebe replikacije korišćeno je virtuelno okruženje (Virtual Machine - VM) koje pruža CERN Open Data platforma. Ovo okruženje je posebno prilagođeno za edukativne i istraživačke svrhe jer uključuje sve neophodne komponente kao što su ROOT biblioteka i CMSSW, čime se obezbeđuje doslednost sa originalnim uslovima analize. Virtuelna mašina takođe omogućava pristup i obradu velikih količina podataka bez dodatne konfiguracije lokalnog sistema.

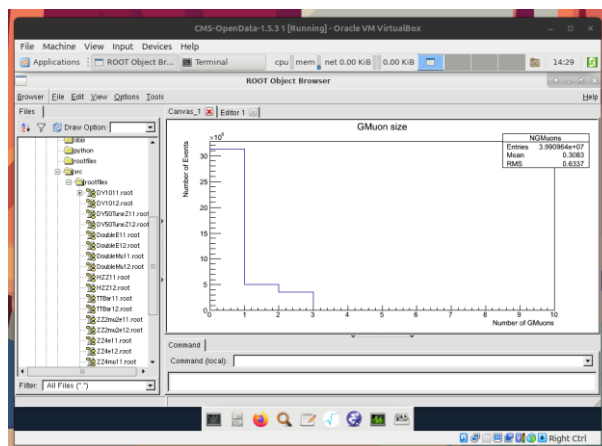


Slika 6. VirtualBox sa CMS VM image-om



Slika 7. Pokretanje CSM virtualne mašine

U okviru virtuelne mašine, pripremljen je novi radni direktorijum u koji su preuzeti svi relevantni ROOT fajlovi sa eksperimentalnim i simulacionim podacima. Da bi analiza bila uspešno izvršena, prethodno je korigovan fajl sa listom adresa, tako da svi linkovi koriste bezbedan protokol (HyperText Transfer Protocol Secure - https). Ova korekcija je neophodna kako bi podaci mogli da budu bezbedno preuzeti sa mreže.



Slika 8. Skinuti ROOT fajlovi

Nakon što su svi fajlovi preuzeti, korišćen je makro kod M4Lnormdatall.cc, koji implementira kompletnu logiku za generisanje složenog histograma. Ovaj kod je dizajniran da učitava ROOT fajlove za različite kanale raspada i potom kreira skalirane histograme za svaku kategoriju: pozadina (ZZ, TTBar, Z/γ^*+X), signal ($H \rightarrow ZZ \rightarrow 4l$) i eksperimentalni podaci (DoubleMu, DoubleE).

Listing 1. Koraci na komandnoj liniji za drugi nivo replikacije

```
→ #postavljanje okruženja
→ cmsrel CMSSW_5_3_32
→ cd CMSSW_5_3_32/src
→ cmsenv
→
→ #nova datoteka
→ mkdir rootfiles
→ cd rootfiles
→
→ #dohvatanje ROOT fajlova
→ wget https://opendata.web.cern.ch/record/5501/files/rootfilelist.txt
→ wget -i rootfilelist.txt
→
→ #preuzimanje i pokretanje C++ makroa
→ wget https://opendata.web.cern.ch/record/5500/files/M4Lnormdatall.cc
→ root -l M4Lnormdatall.cc
```

Makro M4Lnormdatall.cc započinje definisanjem ulaznih parametara neophodnih za korektno skaliranje i interpretaciju podataka iz ROOT fajlova. Ti parametri obuhvataju:

Integrisane luminoznosti za godine 2011. i 2012, izražene u fb^{-1} . Ovi podaci predstavljaju ukupnu količinu sudara zabeleženih tokom relevantnih perioda i ključni su za normalizaciju rezultata.

Teorijske preseke (σ) za sve relevantne procese (npr. $ZZ \rightarrow 4l$, TTBar, DY), koji predstavljaju verovatnoću da se određeni proces dogodi u datom proton-proton sudaru.

Faktore skaliranja (SF) koji se primenjuju na pojedine Monte Karlo simulacije u cilju dodatnog podešavanja rezultata, obično zbog poznatih razlika u efikasnosti detektora ili statističkih odstupanja.

Ukupan broj generisanih događaja u svakom simulacionom fajlu, kako bi se podaci mogli skalirati na realne uslove u eksperimentu.

Nakon definisanja ovih parametara, histogrami se kreiraju za svaki kanal ponaosob. Histogram predstavlja distribuciju mase četiri leptoni ($m_{4\ell}$) za svaki proces. Za sve simulirane podatke (signal i pozadine), koristi se skaliranje prema sledećoj formuli:

$$\omega = \frac{\mathcal{L} \cdot \sigma \cdot SF}{N_{događaja}}$$

Formula 1. Formula za skaliranje podataka

Gde je:

\mathcal{L} – integrisana luminoznost,

σ – teorijski presek,

SF – faktor skaliranja (ako postoji),

$N_{događaja}$ – broj generisanih događaja u Monte Karlo uzorku.

Ova procedura osigurava da se broj događaja iz simulacije pravilno prilagodi količini podataka iz stvarnog eksperimenta. Za signalni kanal ($H \rightarrow ZZ \rightarrow 4\ell$), koristi se unapred skalirani presek, pa dodatni faktor skaliranja nije potreban.

Za eksperimentalne podatke (dobijene direktnim merenjem pomoću DoubleMu i DoubleE detektorskih kanala), histogrami se učitavaju bez skaliranja jer predstavljaju stvaran broj detektovanih događaja i već su normalizovani na uslove akvizicije.

Kreirani histogrami se zatim svrstavaju u četiri glavne kategorije:

- Pozadinski doprinosi iz $ZZ \rightarrow 4\ell$ procesa, koji se formiraju kombinovanjem raspada četiri miona, četiri elektrona i kombinovanih kanala ($2\mu 2e$) iz 2011. i 2012. godine.
- Signalni doprinos Higsovog bozona, koji se prikazuje kao sumirani doprinos raspada $H \rightarrow ZZ \rightarrow 4\mu$, $4e$, i $2\mu 2e$.
- Dodatne pozadine, koje uključuju doprinos iz procesa $Z/\gamma^* + X$ (tzv. Drell-Yan produkcija) i $T\bar{T}$, oba značajna u regionima niskih i srednjih masa.
- Eksperimentalni podaci, koji predstavljaju sumirane merenja iz dva eksperimentalna perioda i više kanala akvizicije.

Za finalnu vizualizaciju koristi se ROOT klasa THStack, koja omogućava slojevito prikazivanje više histograma u okviru istog dijagrama. Svaki proces se razlikuje vizuelno:

- Pozadine su prikazane kao ispunjeni histogrami različitih boja (ZZ — plava, DY — zelena, $T\bar{T}$ — siva),
- Signal Higsovog bozona kao crvena linija,
- Eksperimentalni podaci kao crne tačke sa greškama.

Dijagram uključuje i grafički naslov sa informacijama o energiji sudara ($s = 7$ i 8 TeV) i ukupnim luminoznostima, kao i oznaku "CMS Open Data". U donjem desnom uglu postavljena je i legenda sa objašnjenjima za svaki vizuelni element.

Listing 2. Primer obrade podataka sa parametrima

```
// Input file directory
string inDir = ".";
...
// Name of the input file for MC
string inFileZZ4mu12 = "ZZ4mu12.root";
...
// Luminosity 2012 and 2011
Double_t lumi12 = 11580.;
...
// MC cross section
Double_t xsecZZ412 = 0.077;
...
// Scale factor
Double_t sfZZ = 1.386;
...
// No. of event
Int_t nevtZZ4mu12 = 1499064;
...
// ZZ -> 4mu
TFile *f2 = new TFile((inDir + inFileZZ4mu12).c_str());
TH1D *ZZto4mu12 = (TH1D*) f2->Get("demo/mass4mu_8TeV_low")->Clone();
ZZto4mu12->Scale((lumi12 * xsecZZ412 * sfZZ) / nevtZZ4mu12);
// (data Lumi * xsec * scale factor) / no.of event b4 any cut
```

Ovako strukturisana obrada omogućava jasno poređenje između teorijski predviđenih procesa i eksperimentalnih rezultata, pri čemu je vizuelno evidentan doprinos Higsovog bozona u formi izraženog vrha oko mase od 125 GeV. Ovaj pristup obezbeđuje visok stepen transparentnosti i naučne ponovljivosti, što je jedan od osnovnih ciljeva ove replikacione analize.

Dobijeni dijagram mase četiri leptonu u potpunosti odgovara referentnom prikazu iz originalne CMS publikacije (grafik 3). U njemu je jasno vidljiv vrh oko mase od 125 GeV, što odgovara Higsovoj kontribuciji, dok su pozadinski procesi kvantifikovani sa visokom preciznošću. Vizuelna korespondencija između eksperimentalnih podataka i simulacionih modela dodatno potvrđuje konzistentnost javno objavljenih podataka.

Na taj način, drugi nivo replikacije uspešno pokazuje da je moguće generisati publikovani rezultat korišćenjem javnih izvora i standardnih alata, čime se potvrđuje transparentnost i naučna ponovljivost analize. Ovaj nivo ujedno predstavlja osnovu za složenije analize u narednim fazama replikacije, gde će se koristiti direktniji pristup neobrađenim eksperimentalnim događajima.

4.4. TREĆI NIVO REPLIKACIJE

Treći nivo replikacije predstavlja napredniju fazu analize, čiji je cilj generisanje sopstvenih ROOT fajlova na osnovu sirovih eksperimentalnih i simulacionih podataka dostupnih preko CERN Open Data portala. Za razliku od drugog nivoa, gde su korišćeni unapred obrađeni histogrami, ovde korisnik sam učestvuje u obradi podataka i selekciji događaja, čime se dodatno potvrđuje razumevanje analitičkog toka i eksperimentalne metodologije.

U ovoj fazi, generišu se dve ključne grupe podataka:

- Eksperimentalni podaci, koji sadrže jednog izabranog kandidata za Higsov bozon iz 2012. godine, i
- Monte Karlo simulacije, koje predstavljaju Higsov signal sa namerno smanjenom statistikom, u cilju brže obrade i testiranja algoritma.

Obrada se vrši u okruženju identičnom kao u prethodnom nivou (virtuelna mašina sa CMSSW-om), čime se obezbeđuje kontinuitet u pristupu i kompatibilnost sa ranije postavljenim softverskim alatima. Iako je moguće sprovesti i validacione testove na ovom nivou (Test & Validation sekcija), u ovom radu je taj korak preskočen zbog vremenskih ograničenja, ali to ne utiče na osnovnu funkcionalnost replikacije.

U okviru radnog direktorijuma kreira se posebna struktura koja sadrži konfiguracione fajlove potrebne za kompilaciju i izvođenje analize. Ključni deo ovog nivoa predstavlja pokretanje HiggsDemoAnalyzer programa, koji je odgovoran za obradu sirovih podataka i selekciju događaja. Ovaj program, pomoću odgovarajuće Python konfiguracije (demoanalyzer_cfg_level3data.py i demoanalyzer_cfg_level3MC.py), inicira obradu događaja i upisuje rezultate u nove ROOT fajlove:

- DoubleMuParked2012C_10000_Higgs.root – sadrži jednog odabranog kandidata za Higsov bozon iz eksperimentalnih podataka.
- Higgs4L1file.root – sadrži podatke o simuliranim Higsovim raspadima sa ograničenim brojem događaja.

Korišćenje tzv. "JSON validacionog fajla" osigurava da se u analizu uključuju samo oni događaji koji su zabeleženi u optimalnim uslovima rada detektora (svi podsistemi detektora funkcionalni i bez poznatih anomalija). Time se dodatno povećava kvalitet dobijenih rezultata i izbegava uključivanje nepotpunih ili nepreciznih merenja.

Nakon što su novi ROOT fajlovi generisani, oni se premeštaju u direktorijum sa ostalim fajlovima iz drugog nivoa. Tamo se pokreće novi C++ makro M4Lnormdatall_lvl3.cc, koji ima istu funkcionalnost kao prethodni (M4Lnormdatall.cc), ali je dodatno prilagođen da učitava novonastale podatke i istakne specifičnog kandidata za Higsov bozon.

Listing 3. Koraci u komandnoj liniji za treći nivo replikacije

```
→ #koraci za treći nivo
→ #preuzimanje BuildField.xml
→ cd ~/CMSSW_5_3_32/src
→ mkdir -p Demo/DemoAnalyzer
→ cd Demo/DemoAnalyzer
→ wget https://opendata.web.cern.ch/record/5500/files/BuildFile.xml
→
→ #preuzimanje i kompajliranje HiggsDemoAnalyzer.cc
→ mkdir src
→ cd src
→ wget
https://opendata.web.cern.ch/record/5500/files/HiggsDemoAnalyzer.cc
→ cd ..
→ scram b
→
→ #preuzimanje Python fajlova
→ wget
https://opendata.web.cern.ch/record/5500/files/demoanalyzer\_cfg\_level3data.py
```

```

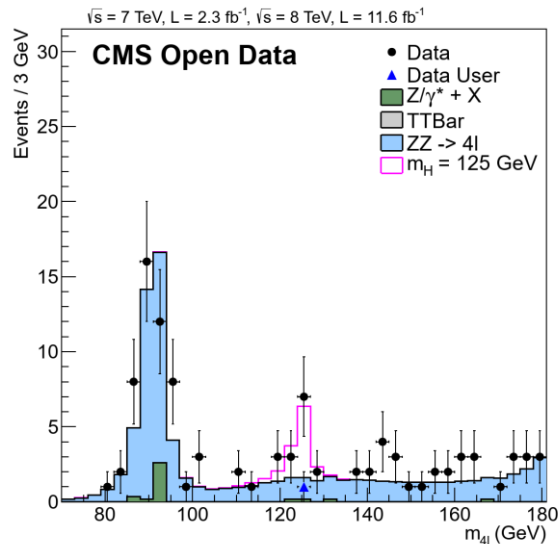
→ wget
  https://opendata.web.cern.ch/record/5500/files/demoanalyzer\_cfg\_level3MC.py
→
→ #nova datoteka za json fajl i za nove root fajlove
→ mkdir datasets
→ cd datasets
→ wget https://opendata.web.cern.ch/record/1002/files/Cert\_190456-208686\_8TeV\_22Jan2013ReReco\_Collisions12\_JSON.txt
→
→ #pokretanje analize podataka
→ cd ..
→ cmsRun demoanalyzer_cfg_level3data.py
→ cmsRun demoanalyzer_cfg_level3MC.py
→
→ #prebacivanje root fajlova u rootfiles
→ mv Higgs4L1file.root ../../rootfiles/
→ mv DoubleMuParked2012C_10000_Higgs.root ../../rootfiles/
→ cd ../../rootfiles
→
→ #preuzimanje i pokretanje c++ makroa
→ wget
  https://opendata.web.cern.ch/record/5500/files/M4Lnormdata11\_lvl3.cc
→ root -l M4Lnormdata11_lvl3.cc

```

Na osnovu obrađenih podataka, makro generiše dijagram mase četiri leptona, pri čemu:

- Higgs kandidat iz eksperimentalnih podataka je označen posebnim markerom (plavi trougao),
- Simulirani podaci i pozadine se prikazuju slojevito, kao i u prethodnom nivou.

Na rezultujućem dijagramu (grafik 4), jasno je uočljiv marker koji označava identifikovanog kandidata za Higsov bozon, tačno na oko 125 GeV, što je u skladu sa poznatim vrednostima iz standardnog modela. Prikaz ovog kandidata u okviru ukupne mase četiri leptona predstavlja važan vizuelni i statistički pokazatelj o mogućnosti posmatranja Higsovog signala u eksperimentalnim podacima.



Grafik 4. Novi dijagram nakon trećeg nivoa

Treći nivo replikacije značajno proširuje prethodnu analizu time što uključuje obradu sirovih podataka i ne oslanja se samo na unapred pripremljene histogram fajlove. Time se ne potvrđuje samo tačnost već objavljenih rezultata, već se i proverava ispravnost i dostupnost javno objavljenih sirovih datasetova, uz primenu osnovnih alata i metoda analize koje koristi CMS kolaboracija. Ova metoda omogućava nezavisnim istraživačima i studentima da praktično rekonstruišu ključne korake u detekciji i potvrdi postojanja Higsove čestice.

4.5. ANALIZA PYTHON I C++ FAJLOVA

Za potrebe trećeg nivoa replikacije korišćena su dva Python fajla:

→ demoanalyzer_cfg_level3data.py (u daljem tekstu data.py),

Listing 4. Podešen JSON validacioni fajl i izvor podataka

```
# define JSON file for 2012 data
goodJSON='/home/cms-
opendata/CMSSW_5_3_32/src/Demo/DemoAnalyzer/datasets/Cert_190456-
208686_8TeV_22Jan2013ReReco_Collisions12_JSON.txt'
myLumis = LumiList.LumiList(filename = goodJSON).getCMSSWString().split(',')
...
# to speed up, pick single example file with 1 nice 2mu2e Higgs candidate
# (9058 events)
process.source = cms.Source("PoolSource",
fileNames =
cms.untracked.vstring('root://eospublic.cern.ch//eos/opendata/cms/Run2012C/DoubleMu
Parked/AOD/22Jan2013-v1/10000/F2878994-766C-E211-8693-E0CB4EA0A939.root')
)
```

→ demoanalyzer_cfg_level3MC.py (u daljem tekstu MC.py).

Listing 5. Podešen izvor podataka

```
# to speed up, read only first file with 7499 events
process.source = cms.Source("PoolSource",
fileNames =
cms.untracked.vstring('root://eospublic.cern.ch//eos/opendata/cms/MonteCarlo2012/Su
mmer12_DR53X/SMHiggsToZZTo4L_M-125_8TeV-powheg15-JHUGenV3-
```



```
pythia6/AODSIM/PU_S10_START53_V19-v1/10000/029D759D-6CD9-E211-B3E2-1CC1DE041FD8.root' )
)
```

Ovi fajlovi predstavljaju konfiguracione skripte za CMS Software Framework (CMSSW) i koriste se za pokretanje C++ analizatora HiggsDemoAnalyzer, koji vrši ekstrakciju relevantnih događaja iz ulaznih datasetova i zapisuje ih u ROOT fajl. Skripte definišu izvore podataka, broj događaja za obradu, strukturu izlaznog fajla i eventualne filtre nad događajima.

Osnovna struktura oba fajla je identična. U oba slučaja, koristi se CMSSW cms.Process("Demo") koji definiše analitički tok. Uključuje se MessageLogger za nadzor izvršenja, a proces se konfiguriše da prikaže sumarne informacije o broju obrađenih događaja.

Broj događaja za obradu postavljen je na -1, što označava da će svi dostupni događaji u fajlu biti procesirani (osim ako nije drugačije definisano).

Ključna komponenta svakog fajla je EDAnalyzer, konkretno instanca HiggsDemoAnalyzer, koja je odgovorna za obradu događaja i selekciju kandidata za Higsov bozon. Analizirani podaci se potom čuvaju pomoću TFileService servisa, koji generiše izlazne ROOT fajlove.

Listing 6. Pozivanje fajla HiggsDemoAnalyzer

```
process.demo = cms.EDAnalyzer('HiggsDemoAnalyzer')
```

Glavna razlika između ova dva fajla jeste izvor podataka, jer data.py koristi eksperimentalne podatke, preuzete sa CMS Open Data portala. Konkretno, koristi se fajl iz DoubleMuParked dataset-a iz 2012. godine, koji sadrži 9058 događaja, uključujući jednog kandidata za Higsov bozon. Ovaj fajl predstavlja unapred izdvojeni deo većeg skupa, što omogućava bržu i efikasniju analizu.

U cilju osiguravanja kvaliteta eksperimentalnih podataka, koristi se i tzv. JSON validacioni fajl (Lumi maska), koji precizira koje periode akvizicije podataka treba uključiti u analizu. Na taj način se isključuju podaci iz perioda kada detektori nisu radili u optimalnim uslovima. JSON fajl se učitava pomoću modula LumiList i primenjuje se nad process.source.lumisToProcess.

MC.py koristi Monte Karlo simulaciju Higsovog bozona u kanalu $H \rightarrow ZZ \rightarrow 4l$. Ovaj dataset sadrži sintetički generisane događaje sa poznatom raspodelom mase, što omogućava preciznu validaciju rada analizatora. U ovom slučaju se JSON validacija ne koristi, jer simulacije nemaju problem sa neispravnim vremenskim periodima.

Takođe, oba fajla eksplicitno navode ulazni fajl putem PoolSource, ali se u komentarima ostavljaju opcije za učitavanje celih datasetova pomoću .txt indeksa fajlova — korisno za šire analize.

Nakon izvršavanja, skripte generišu sledeće ROOT fajlove:

- DoubleMuParked2012C_10000_Higgs.root – rezultat obrade jednog eksperimentalnog događaja u kojem je detektovan potencijalni Higsov kandidat.
- Higgs4L1file.root – skup simuliranih Higsovih raspada, sa smanjenim brojem događaja (7499), koji služi kao referenca za poređenje.

Ovi fajlovi se dalje koriste u četvrtom koraku obrade, u okviru C++ makro koda, gde se kombinuju sa histogramima i ostalim datasetovima kako bi se dobio finalni dijagram raspodele mase četiri leptoni sa jasno obeleženim eksperimentalnim Higs kandidatom.

Makro `HiggsDemoAnalyzer.cc` predstavlja centralni programski modul u trećem nivou replikacije, napisan u C++ jeziku u okviru CMS-ovog softverskog okruženja (CMSSW). Ovaj analizator je odgovoran za obradu sirovih podataka — bilo da su u pitanju eksperimentalni datasetovi ili Monte Karlo simulacije — i za njihovu pretvaranje u fizički značajne histogram dijagrame koji prikazuju raspodele masa leptonskih sistema. Na kraju izvršavanja, generiše se novi ROOT fajl sa brojnim histogramima i (opcionally) ntuple strukturama koje čuvaju kinematičke informacije za rekonstruisane događaje.

Makro započinje sa obimnim uključivanjem standardnih i specifičnih CMS biblioteka, koje omogućavaju pristup podacima o česticama (muoni, elektroni), vertiksima, parametrima traga i uslovima snimanja događaja. Nakon toga, definiše se klasa `HiggsDemoAnalyzer`, koja nasleđuje osnovnu CMSSW klasu `edm::EDAnalyzer`, čime se stiče funkcionalnost za pristup i analizu svakog pojedinačnog događaja iz ulaznog dataset-a.

Unutar klase se nalaze:

- konstruktor i destruktor, koji definišu inicijalizaciju i zatvaranje analiza,
- tri ključne metode: `beginJob()`, `analyze()` i `endJob()`, koje se izvršavaju redom pre, tokom i nakon petlje kroz događaje,
- veliki broj članova klase, među kojima su najznačajniji objekti tipa `TH1D` i `TH2D` (ROOT histogrami), koji predstavljaju raspodele masa, impulsa i drugih fizičkih veličina.

Listing 7. Struktura klase `HiggsDemoAnalyzer`

```
class HiggsDemoAnalyzer: public edm::EDAnalyzer {
public:
    explicit HiggsDemoAnalyzer(const edm::ParameterSet&);
    ~HiggsDemoAnalyzer();
private:
    virtual void beginJob();
    virtual void analyze(const edm::Event&, const edm::EventSetup&);
    virtual void endJob();
    bool providesGoodLumiSection(const edm::Event& iEvent);

    TH1D *h_globalmu_size;
    TH1D *h_recomu_size;
    TH1D *h_e_size;
    ...
}
```

U konstruktoru klase (`HiggsDemoAnalyzer::HiggsDemoAnalyzer`), implementirana je detaljna inicijalizacija svih histogram objekata pomoću ROOT servisa `TFileService`. Ovde se čuvaju histogrami za različite kanale raspada:

- 4μ (četiri miona),
- $4e$ (četiri elektrona),
- $2\mu 2e$ (kombinovani raspad),
- kao i histogrami za kontrolne raspodele (momentum, pseudorapiditet, izolacija, itd.).

Za svaki od navedenih kanala kreirani su histogrami za više varijanti: celokupan opseg mase, fokusirane zone (npr. od 70 do 181 GeV), i sve moguće leptonske kombinacije ($Z12$, $Z34$, itd.). Time se omogućava precizna analiza invarijantne mase leptonskih parova i njihove povezanosti sa signalom Higsovog bozona. Osim fizički značajnih histograma, makro uključuje i niz kontrolnih dijagrama, koji služe za proveru kvaliteta rekonstrukcije čestica.

Konstruktor ne samo da čuva histogram objekte, već i postavlja njihove osnovne parametre:

- broj opsega,
- početne i krajnje vrednosti opsega,
- naslove osa i celokupne nazive.

Takođe se pripremaju svi numerički atributi klase (tipa double, int, TLorentzVector), koji će tokom analize čuvati vrednosti kinematičkih promenljivih za svaki događaj. Većina njih se inicijalizuje na specifične vrednosti kako bi se kasnije lako mogle filtrirati i proveravati na validnost.

Listing 8. Konstruktor klase HiggsDemoAnalyzer

```
HiggsDemoAnalyzer::HiggsDemoAnalyzer(const edm::ParameterSet& iConfig) {  
    // now do what ever initialization is needed  
    edm::Service<TFileService> fs;  
    // *****  
    // book histograms and set axis labels  
    // (called once for initialization)  
    // *****  
    // Global Muon (GM) size  
    h_globalmu_size = fs->make<TH1D>("NGMuons", "GMuon size", 10, 0., 10.);  
    h_globalmu_size->GetXaxis()->SetTitle("Number of GMuons");  
    h_globalmu_size->GetYaxis()->SetTitle("Number of Events");  
    ...  
    ////////////////////////////////// CONTROL PLOTS //////////////////////////////////  
    // Below are the histograms for the control plots  
  
    // Momentum of Global Muon  
    h_p_gmu = fs->make<TH1D>("GM_momentum", "GM momentum", 200, 0., 200.);  
    h_p_gmu->GetXaxis()->SetTitle("Momentum (GeV/c)");  
    h_p_gmu->GetYaxis()->SetTitle("Number of Events");  
    ...  
}
```

Nakon inicijalizacije objekata u konstruktoru, sledeća ključna komponenta makroa HiggsDemoAnalyzer.cc jeste metoda analize, koja se automatski poziva za svaki događaj iz dataset-a. Ova metoda predstavlja centralni operativni segment analize i izvršava se nad svakim pojedinačnim zapisom iz ROOT fajla prosleđenog iz Python konfiguracionih skripti (MC.py i data.py).

U testnim uslovima, broj obrađenih događaja varira u zavisnosti od izvora:

- Dataset iz Monte Karlo simulacije (MC.py) sadrži ukupno 7499 događaja,
- Eksperimentalni dataset (data.py) sadrži 9058 događaja, uključujući jednog validnog kandidata za Higsov bozon.

Na samom početku metode analize, makro zahteva konkretan događaj koji dolazi iz toka podataka (Event). Taj događaj se obrađuje pomoću edm::Handle objekata, koji služe kao ulazne tačke za pristup kolekcijama fizičkih objekata (kao što su mioni, elektroni, vertiksi, i drugi). Svaki Handle se popunjava pozivom iEvent.getByLabel(...), čime se povezuje sa odgovarajućim nazivom kolekcije iz dataset-a.

Nakon što su kolekcije uspešno učitane, sledeći korak je inicijalizacija radnih promenljivih. Ove promenljive — većinom numeričkog tipa (double, int) — služe kao privremeni kontejneri za čuvanje vrednosti kao što su invarijantne mase, transversni impuls, pseudorapiditet, udaljenosti i slično. Inicijalizuju se na vrednosti koje označavaju "nepostojanje" (npr. -9999) kako bi se kasnije mogla izvršiti validacija i filtracija.

Listing 9. Inicijalizacija promenljive u analyze metodi

```
void HiggsDemoAnalyzer::analyze(const edm::Event& iEvent, const edm::EventSetup&
iSetup) {
// *****
// here each relevant event will get analyzed
// *****
nRun = iEvent.run();
nEvt = (iEvent.id()).event(); // iEvent: no class named event()
nLumi = iEvent.luminosityBlock();
...
edm::Handle<reco::TrackCollection> tracks;
iEvent.getByLabel("generalTracks", tracks);
edm::Handle<reco::TrackCollection> gmuons;
iEvent.getByLabel("globalMuons", gmuons);
edm::Handle<reco::MuonCollection> muons;
iEvent.getByLabel("muons", muons);
...
// Initialize variables
eZ12 = -9999.; eZ34 = -9999.; eZ13 = -9999.; eZ24 = -9999.; eZ14 = -9999.; eZ23 = -
9999.; // select largest, init
```

Nakon inicijalizacije sledi glavni blok selekcije podataka, koji se izvodi u okviru for petlji koje iteriraju kroz sve objekte u kolekcijama. Na primer, za svaki mion u događaju proverava se niz kriterijuma:

- da li je rekonstrukcija uspešna,
- da li ima dovoljan broj pogodaka (hits),
- kakva je izolacija (relativna izolacija u odnosu na susedne tragove),
- vrednosti parametara dxy, dz, χ^2 prilagođavanja, itd.

Samo oni objekti (mioni ili elektroni) koji zadovolje sve zadate uslove su selektovani i dalje obrađeni. Selekcija se ne svodi samo na izbacivanje nevalidnih objekata, već uključuje i paralelno čuvanje njihovih indeksa i kinematičkih vrednosti u std::vector strukturama, koje se kasnije koriste za pravljenje leptonskih parova i rekonstrukciju Higsovih raspada.

Uporedo sa filtracijom, makro vrši i punjenje histograma za sve analizirane promenljive. Na primer:

- broj miona i elektrona u događaju,
- njihovi impulsi i mase,
- udaljenosti između traga i primarnog vertex-a.

Listing 10. Validacija i filtriranje podataka

```
// Loop over muons size and select good muons
for (unsigned u = 0; u < muons->size(); u++){
```

```

const reco::Muon &itMuon = (*muons)[u];
math::XYZPoint point(primvtx[0].position());
// select global particle flow muons
// some muons might not have valid track references
if (itMuon.isPFMuon() && itMuon.isPFIsoValid() &&
(itMuon.globalTrack()).isNonnull())
{
    //fill histograms
    h_p_reco->Fill(itMuon.p());
    h_pt_reco_b4->Fill(itMuon.pt());
    h_eta_reco_b4->Fill(itMuon.eta());
    ...
    if (std::abs(SIP3d_mu) < 4. && std::abs((itMuon.globalTrack()->dxy(point)) < 0.5
    && std::abs((itMuon.globalTrack()->dz(point)) < 1. && relPFIso_mu < 0.4){
        if (itMuon.pt() > 5. && std::abs(itMuon.eta()) < 2.4){
            vIdPtmu.push_back( std::make_pair(u, itMuon.pt()) );
        }
    }
} // end of if (itMuon.isPFMuon()).....
} // end muons loop

```

Nakon što su kolekcije čestica očišćene i selektovane prema strogo definisanim kriterijumima kvaliteta, izvršava se finalna faza analize u okviru metode analyze. Ova faza obuhvata fizikalne kalkulacije na osnovu prethodno izdvojenih miona i elektrona i formiranje leptonskih parova koji mogu predstavljati potencijalne raspade bozona Z i, u konačnici, Higgsog bozona.

Postupak rekonstrukcije se primenjuje paralelno i simetrično za oba tipa leptona — mione i elektrone. Prvo se kombinuju selektovane čestice u parove i proverava se njihova ukupna električna naelektrisanost. Ako je zbir naelektrisanja para jednak nuli, što je neophodan uslov za validan leptonski par iz Z bozona, tada se pristupa izračunavanju njihove invarijantne mase putem sumiranja njihovih četvorovektora (TLorentzVector objekti).

Za analizu raspada tipa $Z \rightarrow 2\mu$ i $Z \rightarrow 2e$, uzimaju se dva najkvalitetnija miona ili elektrona (na osnovu kinematičkih parametara) i proverava se da li mogu da čine validan Z kandidat. U slučaju raspada $ZZ \rightarrow 4l$ (gde je $l = e$ ili μ), uzima se kombinacija četiri leptona i razmatraju se sve moguće kombinacije parova. Kombinacije se formiraju tako da se obezbedi nulta ukupna naelektrisanost svakog para, što je preduslov za konzistentnu interpretaciju kao dvostruki raspad Z bozona.

Na osnovu dobijenih leptonskih parova, makro izračunava:

- individualne mase parova (Z_1, Z_2),
- ukupnu masu četvorčestičnog sistema (m_{4l}),
- impulse i energiju svakog sistema.

Ove veličine se koriste za dodatnu selekciju kandidata i popunjavanje specifičnih histogram dijagrama: massZ1, massZ2, mass4l, kao i pomoćne dijagrame koji prikazuju distribucije udaljenosti između leptona, relativne uglove, i slične parametre.

Kao ilustrativan primer može se navesti slučaj rekonstrukcije raspada $Z \rightarrow 2e$. Pronađeni par elektrona sa suprotnim naelektrisanjem i odgovarajućom invarijantnom masom (bliskom 91 GeV, što je masa Z bozona), koristi se za popunjavanje histograma massZeLe. Slično se postupa sa mionima (massZmu) i kombinovanim kanalima (mass2e2mu, mass4mu, mass4e).

Dodatna pažnja se posvećuje tome da se:

- izbegne dupliranje kandidata,

- ne uzimaju iste čestice u više parova,
- i po potrebi primeni kalibracija energija, što može uključivati podešavanje merenih vrednosti na osnovu poznatih sistemskih efekata.

Listing 11. Selekcija kandidata za elektron parove

```
//===== ZTo2Electron start =====//
if (nGoodElectron >= 2)
{
    const reco::GsfElectron &elec1 = (*electrons)[vIdPte.at(0).first];
    const reco::GsfElectron &elec2 = (*electrons)[vIdPte.at(1).first];
    if (elec1.charge() + elec2.charge() == 0){
        for (unsigned i = 0; i < vIdPte.size(); i++){
            // These pT and eta are filled after all the cuts
            // access directly .second as the second pair is already pT
            h_pt_e_after_Zto2e->Fill(vIdPte.at(i).second);
            h_eta_e_after_Zto2e->
Fill(((((*electrons)[vIdPte.at(i).first]).superCluster()->eta()));
        }
        s1 = sqrt(((elec1.p()) * (elec1.p()) + sqme) * ((elec2.p()) * (elec2.p()) +
sqme));
        s2 = elec1.px() * elec2.px() + elec1.py() * elec2.py() + elec1.pz() *
elec2.pz();
        s = sqrt(2.0 * (sqme + (s1 - s2)));
        h_mZ_2e->Fill(s);
    }
}
//===== ZTo2Electron end =====//
```

Nakon završetka svih kalkulacija i popunjavanja histograma, izvršavanje metode analize se završava. S obzirom na to da se ova metoda poziva za svaki događaj pojedinačno, završetkom njenog rada završava se i obrada celog dataset-a.

Završni histogrami, kreirani kroz sve iteracije, se automatski upisuju u ROOT fajl definisan u Python konfiguraciji, čime se kompletira rad ovog C++ makroa. Dobijeni histogrami se dalje koriste za vizuelnu inspekciju i interpretaciju rezultata u kontekstu potvrde postojanja Higsovog bozona.

Makro M4Lnormdatall_lvl3.cc predstavlja završni korak u trećem nivou replikacije. Njegova osnovna funkcija je da učitava sve histogram podatke — kako iz prethodno pripremljenih ROOT fajlova, tako i iz novog korisnički generisanog ulaza — i da ih kombinuje u jedan integrisani dijagram koji prikazuje raspodelu invarijantne mase četiri leptona. Struktura makroa je funkcionalno slična prethodno korišćenom M4Lnormdatall.cc iz drugog nivoa, ali sadrži važne proširene komponente koje omogućavaju uključivanje dodatnih izvora podataka.

U ovom makrou, kao ulazni fajl za simulaciju Higsovog signala iz 2012. godine koristi se novokreirani fajl Higgs4L1file.root, koji je rezultat rada HiggsDemoAnalyzer.cc iz trećeg nivoa. Ovaj fajl sadrži sintetički generisane $H \rightarrow ZZ \rightarrow 4l$ događaje sa smanjenim brojem statističkih uzoraka, što je korisno za testiranje funkcionalnosti analize i bržu obradu.

Dodatno, u makro je uključen i drugi korisnički generisani fajl: DoubleMuParked2012C_10000_Higgs.root. Ovaj fajl sadrži jedan eksperimentalno detektovani kandidat za Higsov bozon, koji je jasno obeležen u finalnom histogramu posebnim markerom (plavi trougao). U tom smislu, makro sada operiše ne samo nad standardnim Monte Karlo simulacijama i zvaničnim eksperimentalnim podacima, već i nad novim korisničkim izvodima, čime se potvrđuje validnost analize u realnim uslovima.

Makro vrši sledeće korake:

- Učitavanje histograma iz različitih ROOT fajlova — za sve relevantne procese
- Skaliranje histograma na osnovu poznatih vrednosti preseka, luminoznosti i broja simuliranih događaja. Za Higgs signal koristi se unapred skalirani presek bez dodatnog faktora korekcije.
- Spajanje histogram komponenti pomoću ROOT klase TStack, kojom se formira slojeviti prikaz različitih doprinosa u ukupnom histogramu.
- Vizuelno formatiranje: dodaju se boje, legende, podešava se veličina markera i osa, postavlja naslov ($\sqrt{s} = 7 \text{ TeV}$, $L = 2.3 \text{ fb}^{-1}$, $\sqrt{s} = 8 \text{ TeV}$, $L = 11.6 \text{ fb}^{-1}$), i unosi CMS oznaka.

Listing 12. Spajanje histograma

```
// Replace the input file for Higgs MC 2012
// with user's generated Higgs input file level 3
string inFileHZZ12 = "Higgs4L1file.root";
// Add the input file for data
// with user's generated data input file level 3
string inFileUser = "DoubleMuParked2012C_10000_Higgs.root";
...
//////////////////// HISTOGRAM 1 DATA FILE FROM USER //////////////////////
f3 = TFile::Open((inDir + inFileUser).c_str());
TH1D *DouMuser = (TH1D*) f3->Get("demo/mass2mu2e_8TeV_low")->Clone();
DouMuser->SetMarkerColor(kBlue);
DouMuser->SetMarkerStyle(22);
DouMuser->SetMarkerSize(2.0);
DouMuser->SetLineColor(kBlack);
DouMuser->SetLineWidth(1);
```

Finalni dijagram (grafik 4) prikazuje:

- Pozadinske procese kao obojene slojeve (plavi, sivi, zeleni),
- Signal Higgsvog bozona kao crvenu liniju,
- Zbirne eksperimentalne podatke kao crne tačke (CMS zvanični podaci),
- Korisnički Higs kandidat kao plavi trougao, pozicioniran tačno oko 125 GeV, što odgovara poznatoj masi Higsove čestice.

Ovaj pristup omogućava da se korisnički podaci ravnopravno uključe u globalnu analizu, sa jasno vidljivim poređenjem u odnosu na poznate teorijske i eksperimentalne rezultate. Time se potvrđuje uspešnost ne samo tehničke implementacije replikacije, već i sposobnost nezavisnog uvida u eksperimentalne tragove postojanja Higgsvog bozona.

5. ČETVRTI NIVO REPLIKACIJE POMOĆU PARALELIZACIJE NA GOOGLE CLOUD PLATFORMI

Četvrti nivo replikacije predstavlja najzahtevniji korak u ovoj analizi, jer podrazumeva rad sa velikim obimom javno dostupnih eksperimentalnih podataka sa CERN Open Data portala. Za razliku od trećeg nivoa, gde je obrađen samo jedan izabrani kandidat za Higsov bozon zajedno sa manjim uzorkom Monte Carlo simulacija, u ovom nivou se pristupa datasetu koji obuhvata stotine miliona događaja.

U uputstvima je naglašeno da se ne koristi kompletan eksperimentalni uzorak koji je CMS kolaboracija analizirala u originalnoj publikaciji, već redukovani dataset, dostupan javnosti u skladu sa CMS Open Data politikom. Ovi podaci čine približno 50% ukupne statistike i predstavljaju tzv. legacy verzije skupova, koje se blago razlikuju od onih u publikaciji zbog kasnijih poboljšanja u kalibraciji i rekonstrukciji. Iako jednostavniji, ovakvi skupovi su korišćeni i u mnogim kasnijim CMS radovima i omogućavaju da se, u duhu originalne analize, kvalitativno prikaže signal Higsovog bozona.

Tehnički izazov ovog nivoa ogleda se u obimu obrade: procesiranje gotovo 350 miliona događaja uz pomoć HiggsDemoAnalyzer makroa predstavlja operaciju koja bi na jednoj virtuelnoj mašini trajala više od mesec dana neprekidnog rada. Kako bi se analiza učinila izvodljivom u realnom vremenskom okviru, bilo je neophodno sprovesti paralelizaciju obrade podataka, tj. podelu dataset-a na više segmenata i njihovu istovremenu obradu.

Za realizaciju paralelizacije korišćena je Google Cloud Platform (GCP), tačnije servis Compute Engine, koji omogućava pokretanje i upravljanje velikim brojem virtuelnih mašina. Na ovaj način kreirano je skalabilno okruženje za masovnu obradu podataka, što je omogućilo da se četvrti nivo replikacije sprovede u skladu sa ciljevima istraživanja.

5.1 PRINCIPI PARALELIZACIJE I PRIMENA U PROJEKTU

Paralelizacija predstavlja osnovni princip savremenog računarskog pristupa velikim problemima. Suština metode ogleda se u tome da se složen zadatak deli na niz manjih i nezavisnih potproblema, koji se zatim obrađuju istovremeno na više procesorskih jedinica ili virtuelnih instanci. Time se skraćuje ukupno vreme izvršavanja, jer više resursa paralelno radi na različitim segmentima istog posla. Ovaj princip je naročito značajan u oblastima gde je obim podataka ogroman i gde bi sekvencijalna obrada bila praktično neizvodljiva u prihvatljivom vremenskom okviru. [17]

U naučnim disciplinama, a posebno u eksperimentalnoj fizici čestica, paralelizacija je postala nezaobilazna metoda. Analize podataka iz eksperimenata poput CMS-a na Velikom hadronskom sudaraču obuhvataju stotine miliona događaja, što daleko prevazilazi kapacitete jedne mašine u razumnom vremenskom periodu. Bez paralelnog izvršavanja, obrada tako velikog broja događaja trajala bi nedeljama ili mesecima, što ograničava pravovremenost naučnih rezultata i efikasnost istraživanja.

Posebna pogodnost podataka iz CMS eksperimenta leži u njihovoj prirodi. Svaki događaj (event) u detektoru predstavlja nezavisan zapis o pojedinačnom sudaru protona. Zbog ove nezavisnosti, događaji ne zahtevaju međusobnu komunikaciju ili koordinaciju tokom obrade. Drugim rečima, analitički tok za jedan događaj ne zavisi od rezultata drugog događaja. To omogućava tzv. podatkovni paralelizam, pri kojem se isti analizator izvršava na više mašina, dok svaka od njih obrađuje svoj zasebni deo skupa podataka.

Jedno od osnovnih ograničenja paralelizacije opisuje Amdahlov zakon. Njegova suština je da ukupno ubrzanje sistema zavisi od dela posla koji se ne može paralelizovati. Ako makar mali procenat koda ostane serijski i mora da se izvrši na jednom procesoru, taj deo postaje usko grlo i određuje gornju granicu performansi. Čak i kada je dominantni deo obrade podeljen na više mašina, preostali serijski segment ograničava ukupni dobitak. Na primer, ako serijski deo

čini svega 5% ukupne obrade, maksimalno ubrzanje ne može premašiti faktor od 20, bez obzira na broj dostupnih procesora. U kontekstu ovog rada, serijski segmenti obuhvataju inicijalizaciju softverskog okruženja, učitavanje i otvaranje ROOT fajlova, kao i završno spajanje histograma iz različitih instanci. Ovi koraci nisu dominantni u ukupnom vremenu, ali njihovo prisustvo znači da povećanje broja virtuelnih mašina daje sve manji marginalni dobitak. Amdahlov zakon tako naglašava da beskonačno ubrzanje nije moguće i da postoji praktična granica skalabilnosti. [17]

Nasuprot tome, Gustafsonov zakon pruža drugačiju perspektivu, polazeći od ideje da se dobitak od paralelizacije najbolje sagledava kada se obim problema povećava. Umesto da se posmatra koliko puta brže može da se reši isti zadatak, fokus je na tome koliko veći zadatak može da se obradi u istom vremenskom okviru ako se poveća broj resursa. Kada se problem proporcionalno uveća sa brojem procesora, serijski deo ostaje približno konstantan, dok paralelni deo raste, čime se povećava ukupna efikasnost. U kontekstu ovog rada to znači da paralelizacija ne omogućava samo kraće vreme obrade, već i proširivanje analize na veće skupove podataka. Na taj način, dataset koji sadrži stotine miliona događaja može se obraditi u prihvatljivom vremenskom okviru, dok bi sekvencijalni pristup zahtevao nedelje ili mesece rada. Gustafsonov zakon time naglašava praktičnu vrednost skaliranja, naročito kod zadataka sa izraženim podatkovnim paralelizmom. [17]

Praktična realizacija principa paralelizacije u ovom radu zasniva se na podeli ukupnog dataset-a na manje segmente i njihovoj nezavisnoj obradi. Svaka virtuelna mašina pokreće isti analizator (HiggsDemoAnalyzer) nad dodeljenim podskupom događaja, čime se ostvaruje takozvani Single Program, Multiple Data (SPMD) pristup. U ovom modelu svi procesi izvršavaju isti kod, ali nad različitim podacima, što u potpunosti odgovara prirodi problema u analizi podataka visokoenergetske fizike.

Budući da su pojedinačni događaji međusobno nezavisni, nema potrebe za složenom komunikacijom ili sinhronizacijom između virtuelnih mašina tokom obrade. Time se izbegavaju režijski troškovi karakteristični za paralelne sisteme sa čestom razmenom poruka i postiže se visoka efikasnost izvršavanja. Jedina faza koja zahteva objedinjavanje jeste završni korak, kada se svi lokalno generisani histogrami kombinuju u jedinstveni skup. Ovaj postupak može da se izvede jednostavnim sabiranjem binova histograma, što predstavlja minimalno opterećenje u odnosu na celokupan obim posla.

Zbog ovih karakteristika, ovakav zadatak se u literaturi klasifikuje kao „embarrassingly parallel“. Ovaj izraz se koristi za probleme kod kojih se obrada lako deli na potpuno nezavisne jedinice rada, sa zanemarljivim zahtevima za komunikaciju među procesima. U takvim slučajevima skaliranje je gotovo linearno: dodavanje dodatnih instanci donosi proporcionalno ubrzanje, sve dok broj događaja dodeljen svakoj instanci ostaje dovoljno velik da se režije ulaza i izlaza podataka ravnomerno rasporede i ne postanu dominantne.

U kontekstu obrade CMS podataka, ovakav pristup omogućava efikasnu analizu i datasetova sa stotinama miliona događaja. Problem se svodi na podelu posla na dovoljno velike blokove podataka po virtuelnoj mašini, tako da svaka instanca maksimalno iskoristi dostupne resurse, a završna faza objedinjavanja rezultata ostaje jednostavna i vremenski zanemarljiva. Ovim se postiže ravnoteža između teorijskih principa paralelizacije i praktičnih zahteva za obradu velikih naučnih datasetova.

5.2 GOOGLE CLOUD PLATFORMA

Google Cloud Platform (GCP) predstavlja integrisani ekosistem servisa koji omogućava izvođenje širokog spektra računarskih zadataka u okruženju oblaka. U osnovi, GCP obezbeđuje infrastrukturu visokih performansi kojom se mogu razvijati, pokretati i skalirati aplikacije, skladištiti i analizirati podaci, kao i izvoditi kompleksni izračuni. Time se korisnicima pruža mogućnost da pristupe resursima koji su inače dostupni samo u okviru velikih data centara, ali bez potrebe za sopstvenim hardverom i infrastrukturnim održavanjem. [18]

Jedan od najvažnijih servisa u okviru GCP-a je Compute Engine, namenjen pokretanju virtuelnih mašina (VM instanci) na Google-ovoj globalnoj infrastrukturi. Compute Engine omogućava fleksibilan izbor tipova mašina (predefinisane ili prilagođene konfiguracije), broja procesorskih jezgara, količine memorije, kao i tipova skladištenja. Ova raznovrsnost čini Compute Engine pogodnim za veoma različite scenarije, od jednostavnih razvojnih okruženja do naučnih projekata koji zahtevaju obradu podataka velikog obima. [18]

Pored tehničkih performansi, GCP se odlikuje i fleksibilnim modelom naplate, poznatim kao pay-as-you-go. Ovaj model podrazumeva da korisnici plaćaju samo onoliko resursa koliko zaista koriste, bez inicijalnog ulaganja u fizičku infrastrukturu. Dodatno, GCP automatski primenjuje popuste za kontinuiranu upotrebu (sustained use discounts), čime se smanjuju troškovi za duže radne zadatke. Postoji i mogućnost korišćenja predplaćenih opcija ili rezervisanih resursa, što pruža dodatne pogodnosti korisnicima sa predvidljivim potrebama i dugoročnim planovima. [18]

Kombinacija skalabilnosti i fleksibilnosti čini GCP pogodnim za istraživačke projekte. Ona omogućava da se eksperimenti sa ogromnim skupovima podataka, poput onih iz visokoenergetske fizike, realizuju efikasno, bez potrebe za skupom i teško održivom lokalnom infrastrukturom.

U okviru ovog rada Google Cloud Platform se koristi kao infrastrukturna osnova za izvođenje paralelne analize podataka iz CMS eksperimenta. Ključni servis je Google Compute Engine, koji omogućava pokretanje više virtuelnih mašina (VM instanci) sa identičnim softverskim okruženjem. Svaka instanca pokreće isti analizator (HiggsDemoAnalyzer), ali obrađuje različit podskup događaja. Na ovaj način realizovan je model Single Program, Multiple Data (SPMD), gde se isti program izvršava nad različitim segmentima dataset-a.

Prvi korak obuhvata raspodelu podataka. Ukupni dataset se deli na blokove koji se dodeljuju pojedinačnim instancama. Ovakav pristup obezbeđuje balans opterećenja i omogućava da svaka instanca obrađuje dovoljan broj događaja kako bi se amortizovali režijski troškovi pokretanja i ulaza/izlaza podataka. Budući da su događaji nezavisni, tokom analize ne postoji potreba za međusobnom komunikacijom između virtuelnih mašina, što značajno smanjuje kompleksnost i povećava efikasnost obrade.

Nakon obrade, svaka instanca generiše lokalne histogram fajlove koji sadrže rezultate. U završnoj fazi, ovi fajlovi se kombinuju u jedan integrisani histogram, čime se dobija konačan prikaz distribucije mase četiri leptona. Ovaj korak predstavlja jedinu tačku objedinjavanja i može se izvesti jednostavnim sabiranjem binova histograma, bez dodatnih komplikacija u komunikaciji među instancama.

Korišćenje GCP-a pruža i mogućnost dinamičkog skaliranja. U zavisnosti od trenutnih potreba, broj aktivnih VM instanci može da se poveća radi obrade većeg obima podataka ili smanji kada je opterećenje manje, čime se optimizuju i performanse i troškovi. Ova fleksibilnost direktno proizlazi iz dizajna GCP servisa, koji korisnicima omogućava da resurse koriste onoliko dugo i u onoj meri koliko im je potrebno, bez dugoročnih infrastrukturnih ulaganja.

5.3 ČETVRTI NIVO REPLIKACIJE

Cilj četvrtog nivoa jeste da se prikaže celokupan proces analize, od obrade ulaznih podataka do dobijanja završnih rezultata u obliku grafikona i histograma. Za razliku od ranijih nivoa, gde se proveravao rad pojedinačnih delova, ovde se naglasak stavlja na povezivanje svih koraka u jedinstvenu celinu. Na taj način potvrđuje se reproducibilnost rezultata i njihova uporedivost sa referentnim analizama.

Ovaj nivo je važan jer pokazuje da alati i procedure korišćeni u prethodnim fazama mogu da funkcionišu zajedno u praksi. Time se potvrđuje da metod nije samo skup izdvojenih eksperimenata, već stabilan i održiv postupak koji se može ponoviti i u drugim istraživačkim okruženjima.

Treba napomenuti da zbog ograničenja u dostupnosti MC fajlova, nije bilo moguće obraditi čitav raspoloživi skup podataka. Umesto toga, podaci koji nedostaju su dopunjeni sa unapred generisanim rezultatima korištenih na prethodnim nivoima replikacije. Na taj način čuva se kontinuitet i obezbeđuje se da metodološki tok rada bude u potpunosti prikazan, iako je obim rezultata smanjen.

5.3.1 PREGLED ULAZNIH PODATAKA

Na četvrtom nivou replikacije koristi se obiman skup podataka organizovan kroz tzv. indexfile-ove. Svaki indexfile predstavlja tekstualnu datoteku u kojoj se nalazi spisak ROOT fajlova. Ti ROOT fajlovi sadrže veliki broj fizičkih događaja, a zajedno čine dataset namenjen analizi. Na taj način indexfile funkcioniše kao svojevrсни katalog koji omogućava da se analitički proces definiše jednom putanjom, umesto da se ručno navodi hiljade pojedinačnih fajlova. Ovakva organizacija podataka posebno je važna na ovom nivou replikacije, gde je cilj da se prikaže metod u punom obimu.

Kao ilustrativan primer može se navesti datoteka `_DoubleElectron_Run2011A-12Oct2013-v1_AOD.txt`, koja sadrži spisak od 1.697 ROOT fajla. Svaki od tih fajlova uključuje određeni broj zabeleženih događaja, a ukupno posmatrano dataset obuhvata 49.693.737 događaja. Obim podataka jasno pokazuje zašto je upotreba indexfile-ova nužna: oni omogućavaju da se ovako veliki skup organizuje na jasan i ponovljiv način, bez potrebe za manuelnim navođenjem svake pojedinačne datoteke.

U okviru konfiguracionih fajlova `demoanalyzer_cfg_level4data.py` i `demoanalyzer_cfg_level4MC.py` podešeno je da se kao ulaz koriste upravo indexfile-ovi. Korišćenjem modula `FileUtils` (`FileUtils.loadListFromFile`) sadržaj indexfile-a se učitava u Python promenljivu, koja zatim postaje ulazni parametar za standardni CMSSW mehanizam `PoolSource`. Na taj način, analiza dobija direktan pristup kompletnom datasetu, a svi događaji se sukcesivno obrađuju bez dodatnih intervencija istraživača.

Listing 13. Podešavanje izvor podataka u demoanalyzer Python fajlovima

```
# *****
# load the data set *
# this example uses one index file of the 2012 DoubleMuParked dataset *
# replace it by the file you wish to treat *
# *****
#
# use the following if you want to run over a full index file
# *** DoubleMuParked2012C_10000 data set (many million events) ***
files2012data = FileUtils.loadListFromFile ('/home/cms-
opendata/CMSSW_5_3_32/src/Demo/DemoAnalyzer/datasets/_DoubleElectron_Run2011A-
12Oct2013-v1_AOD.txt')
process.source = cms.Source("PoolSource",
    fileNames = cms.untracked.vstring(*files2012data
)
)
```

Ovakav način rada omogućava da analiza bude skalabilna i održiva. Skalabilnost se ogleda u činjenici da je moguće raditi sa desetinama miliona događaja, dok održivost dolazi iz same transparentnosti procesa: jasno je dokumentovano koji su fajlovi učitani i na koji način su obrađeni, što omogućava ponovljivost rezultata i njihovu proveru u nezavisnim uslovima. Upravo ova kombinacija obima i metodološke preciznosti daje četvrtom nivou replikacije posebnu važnost u celokupnom radu.

Za pripremu ulaznih podataka u četvrtom nivou korišćen je Google Cloud Shell, komandno okruženje koje funkcioniše kao standardni Linux terminal i direktno je vezano za korisnički nalog u okviru Google Cloud Platforme. U tom okruženju instaliran je program cernopendata-client, alat razvijen za jednostavno preuzimanje i organizovanje podataka sa CERN Open Data portala. Ovaj program omogućava dohvat lokacija ROOT fajlova na osnovu naziva dataset-a ili indexfile-a, čime se proces prikupljanja podataka značajno pojednostavljuje.

Da bi se automatizovalo generisanje spiskova ulaznih fajlova, napisana je Bash skripta generate_indexfiles.sh. Skripta koristi tekstualnu datoteku List_indexfile.txt, u kojoj su unapred definisani svi dataset-ovi koji su relevantni za analizu, uključujući i Monte Karlo simulacije iz 2011. i 2012. godine, kao i eksperimentalne podatke iz istih godina. Za svaki dataset u ovom spisku skripta pokreće komandu cernopendata-client get-file-locations, koja vraća lokacije svih ROOT fajlova dostupnih putem xrootd protokola. Rezultati svake komande čuvaju se u posebnoj tekstualnoj datoteci unutar direktorijuma indexfiles. Nazivi tih izlaznih fajlova generišu se automatski tako što se putanja dataset-a pretvara u jedinstven i „bezbedan“ naziv fajla (zamena kosih crta i razmaka podvlakama).

Listing 14. Skripta za dohvaćanje index fajlova

```
#!/bin/bash

INPUT_FILE="List_indexfile.txt"
OUTPUT_DIR="indexfiles"

# Create output directory if it doesn't exist
mkdir -p "$OUTPUT_DIR"

# Read each line from List_indexfile.txt
while IFS= read -r DATASET_PATH; do
    # Skip empty lines and comments
    [[ -z "$DATASET_PATH" ]] && continue
    [[ ! "$DATASET_PATH" =~ ^/ ]] && continue

    # Make a safe filename from dataset path
    SAFE_NAME=$(echo "$DATASET_PATH" | tr '/' '_' | tr ' ' '_')

    echo "Fetching file locations for: $DATASET_PATH"

    # Run cernopendata-client and save output
    cernopendata-client get-file-locations \
        --title "$DATASET_PATH" \
        --protocol xrootd &> "$OUTPUT_DIR/${SAFE_NAME}.txt"

    echo "Saved to: $OUTPUT_DIR/${SAFE_NAME}.txt"
done < "$INPUT_FILE"
```

Na taj način dobijen je organizovan skup tekstualnih datoteka, od kojih svaka predstavlja indexfile sa kompletnim spiskom ROOT fajlova za odgovarajući dataset. Ovaj postupak omogućio je sistematično i transparentno prikupljanje ulaznih podataka, pri čemu je jasno dokumentovano poreklo svakog fajla i obezbeđena ponovljivost procesa. Rezultat je ujedno i praktična baza ulaznih fajlova koja se dalje koristi u konfiguracionim fajlovima demoanalyzer_cfg_level4data.py i demoanalyzer_cfg_level4MC.py, gde se putem modula FileUtils indexfile-ovi učitavaju i prosleđuju analizi.

Pošto su ulazni skupovi definisani i poznat je prosečni CPU-vreme po događaju iz trećeg nivoa replikacije (0,008732 s/događaju), broj događaja koji jedna VM može da obradi u neprekidnom radu od 24 sata računa se direktno iz odnosa raspoloživog vremena i prosečnog vremena po događaju. Za 24 h (86 400 s) to daje približno 9 894 640 događaja po VM. Na osnovu toga ukupni broj potrebnih VM-ova za dati dataset dobija se deljenjem ukupnog broja događaja u datasetu sa kapacitetom jedne VM u 24 sata.

Primena na dataset DoubleElectron Run2011A (12Oct2013, AOD) sa 49 693 737 događaja pokazuje da je za konzervativno planiranje potrebno približno onoliko VM-ova koliki je ceo deo količnika $49.693.737 / 9.894.640$. Konzervativno zaokruživanje naviše daje 6 VM za sigurno uklapanje u dnevni prozor. U praktičnoj realizaciji ovog nivoa, međutim, korišćeno je 5 VM, što se pokazalo dovoljnim jer je efektivni broj obrađivanih događaja manji: kod realnih podataka deo luminozitetnih intervala biva isključen JSON filtriranjem, a dodatno na ukupno trajanje utiču i sitne varijacije u I/O propusnosti i rasporedu posla.

Kada se isti postupak primeni na sve datasetove uključene u četvrti nivo replikacije, ukupna potrebna infrastruktura iznosi 38 virtuelnih mašina. Od tog broja, 23 VM su potrebne za datasets sa stvarnim podacima, dok je za MC datasetove izračunato da je potrebno 15 VM. U slučaju Monte Karlo skupova dodatni faktor je i sama dostupnost podataka. Kako je naznačeno na CERN Open Data portalu, „dostupan je samo podskup fajlova; kompletan skup se može zatražiti, ali prenos velikih količina na onlajn skladište može potrajati nedeljama ili mesecima“. Zbog toga je stvarni broj događaja za MC bio manji od nominalno prikazanog, pa su se svi ti datasetovi uspešno uklopili u pojedinačne VM instance bez potrebe za dodatnim skaliranjem.

U zbiru, metod zasniva planiranje na mernom proseku iz trećeg nivoa, daje konzervativnu gornju granicu broja VM-ova, a zatim je prilagođava realnim uslovima četvrtog nivoa (filtriranje podataka i dostupnost MC fajlova). Na taj način se postiže balans između robustnosti plana i ekonomičnosti korišćenja resursa.

5.3.2 PRIPREMA VIRTUALNIH MAŠINA ZA OBRADU PODATAKA

Za četvrti nivo korišćen je CernVM (verzija 4.5) za Google Cloud, dostupan preko zvanične stranice (CernVM Appliance). Datoteka sa slikom za GCP najpre se preuzima lokalno, zatim prenosi u Google Cloud Storage (u namenski bucket), a potom se iz te datoteke kreira machine image u okviru Google Clouda. Na osnovu tako pripremljenog image-a pokreću se VM instance, čime se obezbeđuje identično okruženje na svim mašinama. Prilikom kreiranja instanci dodaje se metadata sa korisničkim „user-data“ sadržajem — to je tzv. kontekstualni fajl (ovde: cms-opendata-startup.context) koji automatski podešava okruženje za CMS Open Data analizu odmah po podizanju sistema.

Kontekstualni fajl je strukturisan da se izvršava kroz CernVM-ov amiconfig mehanizam (zato počinje kontrolom da li se izvršava pod amiconfig demonom). U prvim koracima postavlja se CVMFS (CernVM File System) keš na 20 GB i definišu se osnovne CMS promenljive okruženja: CMS_LOCAL_SITE, SCRAM_ARCH=slc6_amd64_gcc472, kao i učitavanje CMS okruženja preko /cvmfs/cms.cern.ch/cmsset_default.sh. Fajl takođe pravi praktičnu komandnu omotač-skriptu cms-shell koja kroz Singularity otvara interaktivni shell sa automatski učitanim CMS okruženjem i vidljivim mount-ovima (CVMFS, EOS i sl.), tako da korisnik odmah dobija konzolu spremnu za rad.

Listing 15. Podešavanje okruženja i CMS Shell omotača u kontekst fajlu

```
#!/bin/sh.before
parents=$(pid=$$; while [ $pid -ne 1 ]; do pid=$(ps -o ppid= -p $pid); cat
/proc/${echo $pid}/cmdline; done)
if ! echo $parents | grep -q amiconfig; then
    echo "Skipping startup script, which should only run under amiconfig daemon"
    exit 0
fi
```

```

# Set CVMFS cache size to 20G
echo "CVMFS_QUOTA_LIMIT=20000 # 20G cache" > /etc/cvmfs/default.local

echo "export CMS_LOCAL_SITE=/etc/cms/SITECONF/T2_UK_London_IC" >
/etc/cvmfs/config.d/cms.cern.ch.local
echo "export SCRAM_ARCH=slc6_amd64_gcc472" > /etc/profile.d/cms.sh
echo "source /cvmfs/cms.cern.ch/cmsset_default.sh" >> /etc/profile.d/cms.sh
echo 'PS1="[Outer Shell \W] "' >> /etc/profile.d/cms.sh
chmod 0755 /etc/profile.d/cms.sh

cat << EOFOUTER > /usr/bin/cms-shell
#!/bin/sh
. . .
EOFOUTER
chmod 0755 /usr/bin/cms-shell

```

Zatim se eliminišu upozorenja klijenta za XRootD postavljanjem protokola na „unix“, a u okviru SITECONF strukture (/etc/cms/SITECONF/T2_UK_London_IC) definišu se mapiranja LFN→PFN ka javnim skladištima (EOS public i relevantni xrootd end-pointovi u UK/EU), kao i fallback lanac koji uključuje globalni redirector. U istom bloku generiše se i site-local-config.xml, koji upućuje analizu na upravo definisane kataloške putanje. Time se VM priprema da transparentno čita fajlove sa CERN-ovih javnih servisa bez dodatne ručne konfiguracije.

Listing 16. Podešavanje mapiranja skladišta podataka u kontekst fajlu

```

# Hack for CMSSW bug (?)
ln -s /cvmfs/cms.cern.ch/SITECONF /etc/cvmfs/SITECONF

# Avoid xroot client warnings
touch /etc/profile.d/xrootd-protocol.sh
echo '
export XrdSecPROTOCOL=unix
' > /etc/profile.d/xrootd-protocol.sh

# Manual SITECONF, ideally at some point part of /cvmfs/cms.cern.ch
mkdir -p /etc/cms/SITECONF/T2_UK_London_IC/{JobConfig,PhEDEx}
ln -s T2_UK_London_IC /etc/cms/SITECONF/local
echo '
. . .
' > /etc/cms/SITECONF/T2_UK_London_IC/JobConfig/site-local-config.xml

```

U delu označenom kao „custom startup script“ fajl se prebacuje na korisnika cms-opendata i iz njegove radne putanje automatizuje preuzimanje i izgradnju analitičkog koda. Kreira se standardna CMSSW struktura, preuzima izvor HiggsDemoAnalyzer.cc, izvršava se scram b kako bi se komponenta izgradila u okviru CMSSW okruženja, a zatim se dopremaju i konfiguracione datoteke za četvrti nivo (demoanalyzer_cfg_level4data.py i demoanalyzer_cfg_level4MC.py). Ujedno se preuzimaju i JSON datoteke za selekciju dobrih luminozitetnih intervala za 7 i 8 TeV podatke, tako da je filter kvaliteta podataka spreman odmah po startu. Na kraju, kontekst blok uključuje i standardnu CernVM konfiguraciju (CVMFS repozitorijume, korisnički nalog, desktop podešavanja) kako bi se obezbedio dosledan interaktivni rad i u grafičkom i u terminalskom režimu.

Listing 17. Podešavanje ličnog startup skripta i CernVM konfiguracije u kontekst fajlu

```
echo "[INFO] Running custom startup script..." | tee /var/log/cernvm-
startup.log

su - cms-opendata <<'EOF' >> /var/log/cernvm-startup.log 2>&1
set -x
cd ~
cmsrel CMSSW_5_3_32
cd CMSSW_5_3_32/src
cmsenv
. . .
mkdir src
cd src
wget https://opendata.web.cern.ch/record/5500/files/HiggsDemoAnalyzer.cc
cd ..
scram b
wget https://opendata.cern.ch/record/5500/files/demoanalyzer_cfg_level4data.py
wget https://opendata.cern.ch/record/5500/files/demoanalyzer_cfg_level4MC.py
. . .
wget https://opendata.web.cern.ch/record/1002/files/Cert_190456-
208686_8TeV_22Jan2013ReReco_Collisions12_JSON.txt
wget https://opendata.cern.ch/record/1001/files/Cert_160404-
180252_7TeV_ReRecoNov08_Collisions11_JSON.txt
EOF

echo "[INFO] Startup script finished." | tee -a /var/log/cernvm-startup.log

exit 0

[cernvm]
repositories=cms.cern.ch,cms-opendata-conddb.cern.ch
shell=/bin/bash
config_url=http://cernvm.cern.ch/config
users=cms-opendata:cms-opendata:password
. . .
```

Ovakva upotreba CernVM image-a i kontekstualnog fajla omogućava da svaka novokreirana instanca bude deterministički uniformna: CVMFS je pravilno podešen, CMS okruženje je aktivno, pristup podacima preko xrootd/EOS je definisan, analitički kod i konfiguracije su preuzeti i izgrađeni, a JSON filteri su dostupni. Rezultat je da su sve VM-ove spremne za pokretanje analize odmah po boot-u, bez dodatnih ručnih intervencija, što je ključno za skaliranje na 38 instanci u okviru dnevnog prozora obrade.

Za automatizovano podizanje CernVM instanci korišćene su Bash skripte koje pozivaju gcloud compute instances create sa unapred definisanim parametrima projekta, zone, slike diska i startup (user-data) konteksta. U skriptama je eksplicitno isključeno dodeljivanje javnih adresa putem opcije --network-interface=...,no-address, čime se poštuje ograničenje na broj eksternih IP (Internet Protocol) adresa i ujedno povećava bezbednost jer VM-ovi nisu direktno izloženi internetu. Pristup instancama se, umesto toga, ostvaruje kroz IAP (Identity-Aware Proxy) tunel (flag --tunnel-through-iap pri korišćenju gcloud compute ssh), što omogućava administraciju bez javne IP adrese. U konkretnim skriptama ovo je tehnički podržano time što

su mrežna sučelja kreirana bez eksternog IP-a i uz odgovarajuće mrežne tagove, dok se „user-data” kontekst (`--metadata-from-file user-data=...`) koristi da bi se okruženje za analizu pripremilo odmah pri bootu (CVMFS, CMSSW, konfiguracije).

Kako VM-ovi nemaju javne IP adrese, potreban je Cloud NAT na nivou regiona/zone kako bi se obezbedio izlazni (egress) saobraćaj ka internetu za neinteraktivne potrebe instanci: preuzimanje paketa i dependencija, pristup CVMFS i xrootd/EOS endpointima (preko posrednih servisa), telemetriju i slične servise. NAT gateway preuzima ulogu “izlazne adrese” za privatne instance, tako da svaka zona u kojoj postoje instance dobija pripadajući NAT (ili centralizovani NAT ako je tako projektovano), što je preduslov da obrada teče bez ručnih izmena mrežnih pravila.

Listing 18. Komande za podešavanje NAT gateway-a

```
→ gcloud compute routers create nat-router \
→   --network=default \
→   --region=europe-central2
→
→ gcloud compute routers nats create nat-config \
→   --router=nat-router \
→   --region=europe-central2 \
→   --nat-all-subnet-ip-ranges \
→   --auto-allocate-nat-external-ips
```

Pored mrežnih ograničenja, poštovana su i kvote infrastrukture. U praksi je bilo moguće podići do ~20 instanci u jednoj regiji (operativno ograničenje/kvota), pa je orkestracija prilagođena tako da se VM-ovi rasporede po zonama/regijama sa aktivnim NAT-om. Dodatno, zbog globalne kvote na broj instanci po projektu (32), obrada je podeljena u dva GCP projekta: jedan namenjen datasetovima sa stvarnim podacima, drugi za MC skupove. U skriptama se to razlikuje kroz vrednosti `--project`, putanju do images (`--create-disk ... image=projects/<project>/global/images/cern-vm-08-10`) i service account identitete, dok su ostali parametri ujednačeni (tip mašine, zona, disk, labele i startup kontekst). Tako, na primer, `DISK_IMAGE` i `SERVICE_ACCOUNT` se razlikuju između verzije za “data” i verzije za “MC”, ali se oba oslanjaju na isti mehanizam user-data i istu komandu kreiranja instance.

Listing 19. Skripta za kreiranje virtualnih mašina

```
#!/bin/bash

VM_NAMES_FILE="vm-names"
#PROJECT="beaming-octagon-462413-s6" #project for data sets
PROJECT="mcsets-analisis-higgs-boson"
#ZONE="europe-west1-b" #First zone, limit 20 instances
ZONE="europe-central2-a"
SERVICE_ACCOUNT="484133232890-compute@developer.gserviceaccount.com"
DISK_IMAGE="projects/beaming-octagon-462413-s6/global/images/cern-vm-08-10"
STARTUP_FILE="./cms-opendata-startup.context"

while read -r VM_NAME; do
    if [ -z "$VM_NAME" ]; then
        continue
    fi

    echo "Dry run: Creating VM $VM_NAME ..."
```



```

gcloud compute instances create "$VM_NAME" \
  --project="$PROJECT" \
  --zone="$ZONE" \
  --machine-type=e2-medium \
  --network-interface=network-tier=PREMIUM,stack-
type=IPV4_ONLY,subnet=default,no-address \
  --maintenance-policy=MIGRATE \
  --provisioning-model=STANDARD \
  --service-account="$SERVICE_ACCOUNT" \
  --
scopes=https://www.googleapis.com/auth/devstorage.read_only,https://www.googlea
pis.com/auth/logging.write,https://www.googleapis.com/auth/monitoring.write,htt
ps://www.googleapis.com/auth/pubsub,https://www.googleapis.com/auth/service.man
agement.readonly,https://www.googleapis.com/auth/servicecontrol,https://www.goo
gleapis.com/auth/trace.append \
  --enable-display-device \
  --tags=http-server,https-server \
  --create-disk=auto-delete=yes,boot=yes,device-name=persistent-disk-
0,image=$DISK_IMAGE,mode=rw,size=20,type=pd-standard \
  --labels=project=cernvm-opendata,vm-name="$VM_NAME" \
  --reservation-affinity=any \
  --metadata-from-file user-data="$STARTUP_FILE"

done < "$VM_NAMES_FILE"

```

U celini, kombinacija privatnih instanci (bez javne IP-adrese), IAP tunelovanja za administraciju i Cloud NAT-a za kontrolisani egress obezbeđuje skalabilno i bezbedno okruženje za pokretanje 38 instanci u okviru dnevnog prozora obrade, uz preciznu reproduktivnost zahvaljujući doslednom CernVM image-u i kontekstualnom fajlu.

Nakon što su VM instance podignute iz pripremljene CernVM slike i konfigurisanog kontekstualnog fajla, sledeći korak bio je dodeljivanje odgovarajućih indexfile-ova. Za svaku instancu unapred je određen dataset koji obrađuje, a pripadajući indexfile je prebačen na tu VM preko gcloud scp komande. Na taj način se osigurava da svaka instanca ima precizan spisak ROOT fajlova nad kojima će izvršiti analizu.

Unutar konfiguracionih datoteka demoanalyzer_cfg_level4data.py i demoanalyzer_cfg_level4MC.py zatim su izvršene prilagodbe tako da koriste upravo taj indexfile. Pored toga, za svaku instancu bilo je potrebno podesiti parametre skipEvents i maxEvents. Prvi parametar omogućava da se preskoči odgovarajući broj događaja, čime se dataset ravnomerno deli među instancama, dok drugi parametar ograničava ukupan broj obrađenih događaja na maksimum od 9 894 640, što predstavlja kapacitet jedne VM za 24 sata rada. Na taj način je svaka instanca dobila svoj segment podataka, a kompletan dataset je obrađen paralelno, bez preklapanja i bez nepotrebnog ponavljanja događaja.

Listing 20. Podešavanje maxEvents i skipEvents parametara

```

# *****
# set the maximum number of events to be processed *
# this number (argument of int32) is to be modified by the user *
# according to need and wish *
# default is preset to -1 (all events) *
# 9894640 events per 24 hour *

```

```
# *****
process.maxEvents = cms.untracked.PSet( input = cms.untracked.int32(9894640) )
...

# *****
# number of events to be skipped (0 by default) *
# *****
process.source.skipEvents = cms.untracked.uint32(3*9894640)
```

Za pokretanje analize korišćena je komanda cmsRun, ali u kombinaciji sa alatima za kontrolu izvršavanja. Analiza je startovana pomoću nohup cmsRun demoanalyzer.py &> naziv_indexfile.report &, čime se postiže dvostruka funkcionalnost: s jedne strane, proces nastavlja sa radom i nakon zatvaranja terminala, a s druge strane, sav izlaz (informacije o toku rada, greške i sažeci) beleži se u poseban report fajl. Na taj način moguće je naknadno pratiti rad svakog posla i imati uvid u to da li je analiza uspešno završena.

Ovim pristupom uspešno je organizovana i pokrenuta obrada podataka na 38 virtuelnih mašina, pri čemu je svaka imala jasno definisan segment posla i kompletno podešeno okruženje. Kombinacija podeljenih indexfile-ova, preciznih parametara u DemoAnalyzer-u i kontrolisanog pokretanja preko nohup komande omogućila je da se celokupna analiza izvede efikasno i reproduktivno.

5.3.3 REZULTATI OBRADE I VREMENSKA ANALIZA

Po završetku izvršavanja analize na svakoj virtuelnoj mašini, prvi zadatak bio je rad sa generisanim report fajlovima. Budući da su oni često veoma veliki, bilo je potrebno smanjiti njihovu veličinu pre prenosa. To je urađeno tako što je iz svakog report-a izdvojeno po 200 linija od početka i 200 linija od kraja, što je omogućilo da se sačuvaju ključne informacije o toku izvršavanja i eventualnim greškama, a da datoteke budu značajno manje i preglednije. Nakon toga report i odgovarajući ROOT fajl kopirani su u lokalno okruženje, čime je obezbeđena centralizovana kontrola i priprema za završne korake.

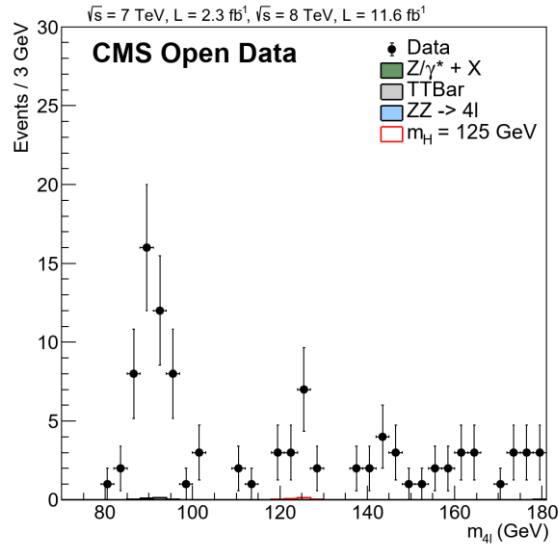
U sledećoj fazi, svi ROOT fajlovi sa pojedinačnih instanci objedinjeni su na finalnoj virtuelnoj mašini korišćenjem alata hadd, koji omogućava spajanje više izlaznih ROOT fajlova u jedan objedinjeni dataset. Na ovaj način dobijeni su kompletni rezultati, spremni za dalju analizu.

Po završetku izvršavanja analize na svakoj virtuelnoj mašini, prvi zadatak bio je rad sa generisanim report fajlovima. Budući da su oni često veoma veliki, njihova veličina je smanjena izdvajanjem po 200 linija sa početka i 200 linija sa kraja. Tako sačuvani izveštaji zadržavaju ključne informacije o izvršavanju i potencijalnim greškama, a istovremeno su pregledniji i praktičniji za prenos. Nakon toga report i odgovarajući ROOT fajl kopirani su u lokalno okruženje, odakle su objedinjeni na finalnoj virtuelnoj mašini.

Objedinjavanje je izvršeno pomoću alata hadd, čime su svi izlazni ROOT fajlovi spojeni u jedan dataset. Nakon toga je u fajlu M4Lnormdatall.cc ažuriran spisak izvora kako bi se omogućilo poređenje različitih kombinacija podataka.

Kao što je već prikazano na grafiku 3 u prethodnom poglavlju, očekivani rezultat histograma mase četiri leptona služi kao referentna vrednost za poređenje. U ovom delu predstavljene su tri varijante dobijene u četvrtom nivou replikacije, koje ilustruju uticaj dostupnosti i porekla podataka na konačni oblik distribucije.

Na grafiku 5 prikazani su rezultati dobijeni korišćenjem isključivo MC podataka generisanih u okviru ovog rada. Budući da, kao što je već napomenuto, MC datasetovi nisu u potpunosti dostupni na javnim portalima, histogram jasno pokazuje praznine i odstupanja koja su posledica nedostatka podataka.



Grafik 5. Rezultat sa generisanim MC podacima

Listing 21. Podešavanje ulaznih fajlova - generisani MC podaci

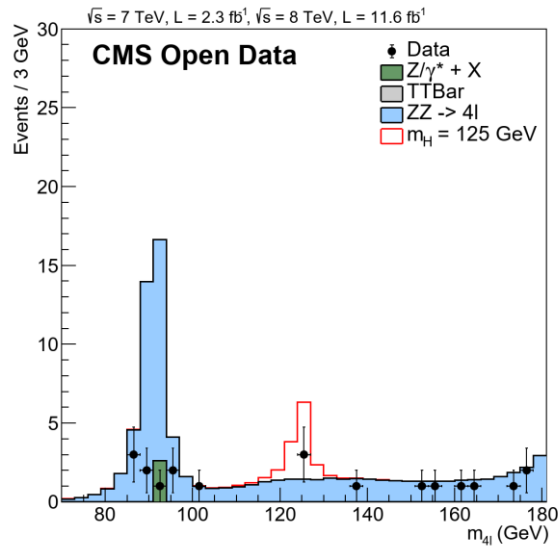
```
// Name of the input file for MC
string inFileZZ4mu12 = "ZZTo4mu_8TeV_12.root"; // "ZZ4mu12.root";
string inFileZZ4e12 = "ZZTo4e_8TeV_12.root"; // "ZZ4e12.root";
string inFileZZ2mu2e12 = "ZZTo2e2mu_8TeV_12.root"; // "ZZ2mu2e12.root";
string inFileZZ4mu11 = "ZZTo4mu_m114_7TeV_11.root"; // "ZZ4mu11.root";
string inFileZZ4e11 = "ZZTo4e_m114_7TeV_11.root"; // "ZZ4e11.root";
string inFileZZ2mu2e11 = "ZZTo2e2mu_m114_7TeV_11.root"; // "ZZ2mu2e11.root";

string inFileHZZ12 = "SMHiggsToZZTo4L_M-125_8TeV_12.root"; // "HZZ12.root";
string inFileHZZ11 = "SMHiggsToZZTo4L_M-125_7TeV_11.root"; // "HZZ11.root";

string inFileTTBar12 = "TTbar_8TeV_12.root"; // "TTBar12.root";
string inFileTTBar11 = "TTTo2L2Nu2B_7TeV_11.root"; // "TTBar11.root";

string inFileDY5012 = "DYJetsToLL_M-50_TuneZ2_12.root"; // "DY50TuneZ12.root";
string inFileDY5011 = "DYJetsToLL_TuneZ2_M-50_11.root"; // "DY50TuneZ11.root";
string inFileDY1012 = "DYJetsToLL_M-10to50_12.root"; // "DY1012.root";
string inFileDY1011 = "DYJetsToLL_M-10To50_11.root"; // "DY1011.root";
```

Grafik 6 U ovoj varijanti korišćeni su MC podaci preuzeti iz prethodnih nivoa replikacije, dok su stvarni podaci namerno uključeni samo delimično, kako bi se pokazalo da takav skup ne može u potpunosti da reprodukuje originalni rezultat. Zbog nepotpunog obuhvata i manje statistike, histogram pokazuje izraženije fluktuacije i lokalna odstupanja u odnosu na referentni oblik sa grafika 3. Takvo ponašanje je očekivano kada su pojedini periodi i kanali nedovoljno pokriveni, a normalizacija MC i realnih podataka nije u potpunosti stabilizovana. Stoga se grafik 6 koristi prvenstveno za ilustraciju uticaja parcijalno dostupnih podataka, a ne za izvođenje kvantitativnih zaključaka.

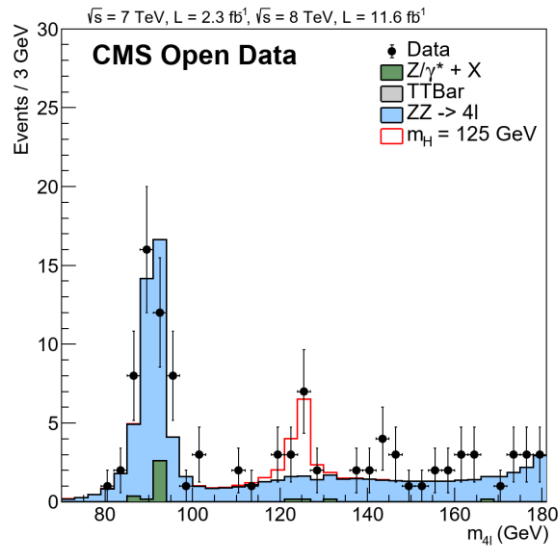


Grafik 6. Rezultat sa delimično uključenim stvarnim podacima

Listing 22. Podešavanje ulaznih fajlova - delimično uključeni stvarni podaci

```
// Name of input file for data
string inFileDouMu12 = "../DoubleMuParked_Run2012B-3_partial.root";
// "DoubleMu12.root";
string inFileDouMu11 = "../DoubleMu_Run2011A-4_partial.root";
// "DoubleMu11.root";
string inFileDouE12 = "DoubleElectron_Run2012_partial.root";
// "DoubleE12.root";
string inFileDouE11 = "DoubleElectron_Run2011.root"; // "DoubleE11.root";
```

Na grafiku 7 prikazana je kombinacija unapred generisanih podataka sa prethodnih nivoa i novi ROOT fajlovi generisani tokom ovog rada. Histogram potvrđuje da su stvarni podaci ispravni, jer nema razlike u odnosu na očekivani rezultat iz grafika 3. Ovo pokazuje da metod ostaje pouzdan i kada se oslanja na parcijalno dostupne MC podatke i ceo obim novo-generisanih realnih podataka.



Grafik 7. Rezultat sa svim generisanim podacima

Listing 23. Podešavanje ulaznih fajlova – svi generisani podaci

```
// Name of input file for data
string inFileDouMu12 = "DoubleMuParked_Run2012.root"; //"DoubleMu12.root";
string inFileDouMu11 = "DoubleMu_Run2011.root"; //"DoubleMu11.root";
string inFileDouE12 = "DoubleElectron_Run2012.root"; //"DoubleE12.root";
string inFileDouE11 = "DoubleElectron_Run2011.root"; //"DoubleE11.root";
```

U sve tri varijante primenjene su odgovarajuće izmene u fajlu M4Lnormdata11.cc, gde su definisani izvori podataka korišćeni u analizi. Time je obezbeđena transparentnost u izboru datasetova i jasno dokumentovan način na koji su grafici dobijeni.

Planirano je da se celokupna obrada podataka završi u roku od 24 sata, na osnovu ranije izmerenog prosečnog vremena obrade po događaju od 0,008732 s. Međutim, tokom praktične realizacije pokazalo se da stvarni rezultati značajno odstupaju od očekivanja.

Dok su pojedine virtuelne mašine postigle planirane performanse i završile obradu u predviđenom roku, kod drugih je prosečno vreme po događaju bilo i duplo ili čak tri puta veće od očekivanog. To je dovelo do toga da je ukupan vremenski okvir za analizu na nekim instancama premašio 24 sata, a u pojedinim slučajevima je obrada trajala i više od 72 sata, čime je plan značajno prekoračen.

Glavni razlog za ovakva odstupanja bile su network operacije, koje su oduzimale najveći deo resursa i vremena. Analiza podataka se u velikoj meri oslanja na čitanje ROOT fajlova preko mrežnih protokola (xrootd, EOS), pa je svako usporenje ili zagušenje u mrežnoj infrastrukturi imalo direktan uticaj na brzinu obrade. Posebno se primećivalo da su virtuelne mašine koje su među prvima puštene u rad uglavnom ispunile očekivanja, dok su one koje su naknadno pokretane bile sve sporije. Najverovatnije je reč o opterećenju sistema i infrastrukture — kako je broj aktivnih instanci rastao, tako su i mrežni resursi postajali zagušeniji, što se reflektovalo na ukupno trajanje obrade.

Na slici 9 prikazani su rezultati u slučaju kada je ostvareno očekivano vreme obrade (oko 0,0087 s po događaju).

```

TrigReport ----- Event Summary -----
TrigReport Events total = 9894640 passed = 9894640 failed = 0

TrigReport ----- Path Summary -----
TrigReport Trig Bit# Run Passed Failed Error Name
TrigReport 1 0 9894640 9894640 0 0 p

TrigReport -----End-Path Summary -----
TrigReport Trig Bit# Run Passed Failed Error Name

TrigReport ----- Modules in Path: p -----
TrigReport Trig Bit# Visited Passed Failed Error Name
TrigReport 1 0 9894640 9894640 0 0 demo

TrigReport ----- Module Summary -----
TrigReport Visited Run Passed Failed Error Name
TrigReport 9894640 9894640 9894640 0 0 demo
TrigReport 9894640 9894640 9894640 0 0 TriggerResults

TimeReport ----- Event Summary ---[sec]----
TimeReport CPU/event = 0.001116 Real/event = 0.008395

```

Slika 9. Rezultati sa očekivanom brzinom obrade ($\sim 0,0087$ s po događaju)

Na slici 10 prikazan je primer dataset-a kod kojeg je prosečno vreme bilo približno duplo u odnosu na planiranu vrednost.

```

TrigReport ----- Event Summary -----
TrigReport Events total = 9894640 passed = 9894640 failed = 0

TrigReport ----- Path Summary -----
TrigReport Trig Bit# Run Passed Failed Error Name
TrigReport 1 0 9894640 9894640 0 0 p

TrigReport -----End-Path Summary -----
TrigReport Trig Bit# Run Passed Failed Error Name

TrigReport ----- Modules in Path: p -----
TrigReport Trig Bit# Visited Passed Failed Error Name
TrigReport 1 0 9894640 9894640 0 0 demo

TrigReport ----- Module Summary -----
TrigReport Visited Run Passed Failed Error Name
TrigReport 9894640 9894640 9894640 0 0 demo
TrigReport 9894640 9894640 9894640 0 0 TriggerResults

TimeReport ----- Event Summary ---[sec]----
TimeReport CPU/event = 0.001918 Real/event = 0.017560

```

Slika 10. Rezultati sa prosečnim vremenom obrade približno duplo u odnosu na očekivano vreme

Na slici 11 prikazan je slučaj gde je vreme obrade bilo približno tri puta veće od očekivanog.

```

TrigReport ----- Event Summary -----
TrigReport Events total = 9894640 passed = 9894640 failed = 0

TrigReport ----- Path Summary -----
TrigReport Trig Bit# Run Passed Failed Error Name
TrigReport 1 0 9894640 9894640 0 0 p

TrigReport -----End-Path Summary -----
TrigReport Trig Bit# Run Passed Failed Error Name

TrigReport ----- Modules in Path: p -----
TrigReport Trig Bit# Visited Passed Failed Error Name
TrigReport 1 0 9894640 9894640 0 0 demo

TrigReport ----- Module Summary -----
TrigReport Visited Run Passed Failed Error Name
TrigReport 9894640 9894640 9894640 0 0 demo
TrigReport 9894640 9894640 9894640 0 0 TriggerResults

TimeReport ----- Event Summary ---[sec]----
TimeReport CPU/event = 0.002396 Real/event = 0.024274

```

Slika 11. Rezultati sa prosečnim vremenom obrade oko tri puta veće od planirane vrednosti

Ova iskustva ukazuju na važnost konzervativnijeg planiranja resursa. Iako je teorijski proračun pokazao da je 38 virtuelnih mašina dovoljno za obradu kompletnog dataset-a u jednom danu, praktična realizacija je otkrila da infrastruktura ima ograničenja koja se ne mogu zanemariti. U budućim replikacijama bilo bi preporučljivo računati sa rezervom — kako u vremenskim procenama, tako i u raspodeli datasetova — kao i predvideti dodatne mehanizme za balansiranje mrežnog opterećenja. Uprkos ovim izazovima, analiza je uspešno završena, ali iskustvo jasno pokazuje da mrežne performanse predstavljaju ključni faktor u obradi velikih otvorenih podataka u cloud okruženju.

Ukupna cena sprovedene obrade, uključujući prethodne pripreme i testiranja, iznosila je preko 500 USD. Iako testne faze nisu zahtevale veliki broj resursa, njihovi troškovi su uključeni u konačan iznos kako bi se dobila potpuna slika. Najveći deo troškova odnosio se na networking i mrežne operacije, što se jasno vidi i na slici 12, gde je grafički prikaz potrošnje po servisima Google Cloud Platforme. Compute Engine i Cloud Storage troškovi su činili manji deo ukupnog budžeta.



Slika 12. Prikaz troškova obrade na Google Cloud Platformi

Ovo ukazuje da mrežna infrastruktura predstavlja ne samo tehničko, već i finansijsko ograničenje u radu sa velikim datasetovima. Međutim, ako su virtuelne mašine optimizovani

za mrežne operacije, to bi moglo da obezbediti bolji odnos performansi i troškova. U budućim replikacijama preporučljivo je ispitati dostupne opcije i izabrati instance specijalizovane za intenzivne mrežne operacije, što bi omogućilo efikasnije korišćenje resursa i smanjenje ukupnih troškova.

6. ZAKLJUČAK

Cilj ovog rada bio je da se prikaže mogućnost replikacije analize raspada Higsovog bozona u četiri leptona ($H \rightarrow ZZ \rightarrow 4l$) korišćenjem javno dostupnih podataka sa CERN Open Data portala. Kroz više nivoa replikacije, postepeno je razvijan metodološki okvir koji je omogućio da se složena analiza sprovede i u nezavisnom istraživačkom okruženju. Poseban značaj imao je četvrti nivo, gde je obim podataka bio dovoljno velik da zahteva korišćenje distribuiranog sistema obrade i pažljivo planiranje raspoloživih resursa. Upravo kroz ovaj nivo jasno je pokazano da replikacija nije samo tehnički izazov, već i organizacioni proces koji zahteva optimizaciju vremena, infrastrukture i budžeta. Na taj način potvrđeno je da principi otvorene nauke mogu biti primenjeni i u praksi, dok se istovremeno razvijaju veštine i metode potrebne za rad sa velikim datasetovima u oblasti fizike visokih energija.

Najvažnije rešenje postavljenog problema odnosilo se na organizaciju obrade datasetova sa desetinama miliona događaja. Empirijski određeno prosečno vreme obrade po događaju omogućilo je proračun kapaciteta jedne virtuelne mašine u okviru 24 sata, što je poslužilo kao osnova za planiranje ukupnog broja instanci. Na taj način obezbeđena je efikasna upotreba 38 virtuelnih mašina, od kojih je 23 bilo namenjeno stvarnim podacima, a 15 Monte Karlo simulacijama. Ova podela pokazala se kao optimalna u uslovima budžetskih i tehničkih ograničenja, jer je omogućila balans između raspoloživih resursa i obima podataka koji je trebalo obraditi. Rezultati dobijeni na ovaj način potvrdili su da je pažljivo planiranje i pravilna raspodela posla ključni faktor za uspešno sprovođenje analize u cloud okruženju, gde se preciznost u proceni resursa direktno odražava na kvalitet i pouzdanost konačnih nalaza.

Realizacija ovog plana oslanjala se na Google Cloud Platformu i korišćenje CernVM image-a, čime je postignuto uniformno okruženje za sve instance. Upotrebom kontekstualnih fajlova i automatizovanih skripti, okruženje je bilo spremno za analizu odmah po pokretanju virtuelnih mašina, što je značajno smanjilo potrebu za ručnim intervencijama i omogućilo reproduktivnost rezultata. Na ovaj način praktično je demonstrirano da je moguće organizovati kompleksnu obradu podataka na osnovu otvorenih resursa i javno dostupnih tehnologija. Pored toga, ovakav pristup pokazao je da se napredni eksperimenti iz oblasti fizike visokih energija mogu uspešno prilagoditi savremenim metodama distribuiranog računanja, pri čemu cloud infrastruktura postaje ključni element u prevazilaženju tehničkih i logističkih ograničenja. Time je potvrđeno da otvorena nauka nije samo koncept transparentnosti, već i efikasan model praktične primene u realnim istraživačkim uslovima.

Poseban izazov predstavljala je nepotpuna dostupnost MC datasetova, što je ograničilo kvalitet i stabilnost pojedinih histograma. Na grafikonu dobijenom isključivo na osnovu dostupnih MC podataka jasno su uočene praznine i fluktuacije, što je posledica nedostatka velikog dela događaja. Da bi se ovaj problem prevazišao, u analizu su uključeni podaci iz prethodnih nivoa replikacije, čime je postignuta stabilnost i konzistentnost. Ovo pokazuje da otvoreni podaci mogu da posluže kao snažna baza za istraživanje, ali i da njihova ograničena dostupnost zahteva dodatnu metodološku prilagodljivost. Uprkos delimičnim prazninama u MC podacima, kombinovanjem postojećih i unapred obrađenih datasetova uspešno je sačuvana pouzdanost rezultata, što potvrđuje da replikacija može dati relevantne zaključke i u uslovima kada podaci nisu potpuni.

Analiza je takođe pokazala da su stvarni podaci dali rezultate koji se potpuno uklapaju u očekivani oblik histograma. Time je potvrđeno da je metod ispravan i pouzdan, a da ograničenja u dostupnosti simulacija ne narušavaju glavne zaključke. Na osnovu poređenja različitih varijanti datasetova može se reći da su ključne karakteristike raspada Higsovog bozona reprodukovane uspešno i u ovom nezavisnom radu.

Rezultati istraživanja potvrđuju nekoliko važnih činjenica. Prvo, demonstrirano je da se složene analize iz oblasti fizike čestica mogu uspešno sprovesti koristeći javno dostupne podatke i relativno ograničenu infrastrukturu. Drugo, pokazano je da planiranje i raspodela resursa igraju ključnu ulogu u ostvarivanju zadatih ciljeva, naročito kada se radi o datasetovima velikih dimenzija. Treće, potvrđeno je da metodološka transparentnost, kroz korišćenje jasno dokumentovanih skripti i automatizovanih procedura

Na osnovu ovih nalaza mogu se formulisati i određene preporuke. Preporučuje se da se pri radu sa CERN Open Data podacima uvek vodi računa o dostupnosti datasetova i da se analize planiraju uzimajući u obzir eventualne nedostatke u obimu podataka. Takođe, preporučuje se veća upotreba automatizacije u radu sa cloud infrastrukturom, jer se time smanjuje mogućnost greške i olakšava kontrola velikog broja instanci.

Preporuke koje proističu iz ovog rada ukazuju na nekoliko pravaca daljeg unapređenja. Najpre, iskustvo pokazuje da je pri radu sa otvorenim podacima potrebno voditi računa o njihovoj dostupnosti i obimu. Nepotpuni skupovi, posebno oni koji se odnose na Monte Karlo simulacije, mogu značajno uticati na stabilnost rezultata, pa je neophodno od početka imati plan kako kombinovati različite izvore podataka i nadoknaditi nedostatke. Jednako važno jeste razvijati visok stepen automatizacije u radu sa cloud infrastrukturom. Upravo taj pristup omogućava pouzdan i ponovljiv tok rada, smanjuje mogućnost ljudskih grešaka i čini upravljanje velikim brojem instanci jednostavnijim i efikasnijim. Konačno, preporučuje se da se metodologija dalje širi na druge kanale raspada Higsovog bozona i na novije skupove podataka iz LHC eksperimenata. Time bi se omogućilo ne samo testiranje konzistentnosti već i proširivanje domena primene otvorenih podataka, što bi doprinelo njihovoj još većoj vrednosti za istraživačku zajednicu.

Ukupno posmatrano, rad je uspešno dao odgovore na postavljena istraživačka pitanja i potvrdio da je replikacija analize Higsovog bozona u četiri leptona moguća i izvan originalne kolaboracije. Ovim se doprinosi daljoj afirmaciji otvorene nauke i pokazuje da principi transparentnosti, dostupnosti i reproduktivnosti nisu samo teorijski ideali, već praktične smernice koje omogućavaju razvoj nauke u savremenom društvu. Time je rad ispunio svoj cilj i dao osnovu za dalja istraživanja i primene u ovoj oblasti.

REFERENCE

- [1] Chatrchyan, S, et al., "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC," *Physics Letters B*, vol. 716, no. 1, pp. 30-61, 2012.
- [2] Chatrchyan, S, et al., "Combined results of searches for the standard model Higgs boson in pp collisions at $\sqrt{s}=7$ TeV," *Physics Letters B*, vol. 710, no. 1, pp. 26-48, 2012.
- [3] Collaboration, C M S, et al., "Search for the Standard Model Higgs Boson in the Decay Channel $H \rightarrow ZZ \rightarrow 4l$ in pp Collisions at $\sqrt{s} = 7$ TeV," *Physical Review Letters*, vol. 108, no. 11, p. 111804, 3 2012.
- [4] Evans, Lyndon and Bryant, Philip, "LHC Machine," *Journal of Instrumentation*, vol. 3, no. 08, pp. S08001-S08001, 2008.
- [5] Kasieczka, Gregor, et al., "The LHC Olympics 2020 a community challenge for anomaly detection in high energy physics," *Reports on Progress in Physics*, vol. 84, no. 12, p. 124201, 2021.
- [6] Griffiths, David, *Introduction to Elementary Particles*, s.l. : John Wiley & Sons, 2008. 9783527618477.
- [7] Close, F E, *Particle Physics : a Very Short Introduction*, s.l. : Oxford University Press, 2004. 9780192804341.
- [8] Itzykson, Claude and Zuber, Jean-Bernard, *Quantum Field Theory*, s.l. : Courier Corporation, 2012. 9780486134697.
- [9] Aad, G, et al., "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC," *Physics Letters B*, vol. 716, no. 1, pp. 1-29, 2012.
- [10] Higgs, Peter W, "Broken Symmetries and the Masses of Gauge Bosons," *Physical Review Letters*, vol. 13, no. 16, pp. 508-509, 1964.
- [11] Englert, F and Brout, R, "Broken Symmetry and the Mass of Gauge Vector Mesons," *Physical Review Letters*, vol. 13, no. 9, pp. 321-323, 1964.
- [12] Cremonesi, Matteo, et al., "Using Big Data Technologies for HEP Analysis," *EPJ Web of Conferences*, vol. 214, no. 06030, 2019.
- [13] Clarke, P, et al., *Big Data in the physical sciences: challenges and opportunities*, 2016.
- [14] Brun, Rene and Rademakers, Fons, "ROOT — An object oriented data analysis framework," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 389, no. 1-2, pp. 81-86, 1997.
- [15] Antcheva, I, et al., "ROOT — A C++ framework for petabyte data storage, statistical analysis and visualization," *Computer Physics Communications*, vol. 180, no. 12, pp. 2499-2512, 2009.
- [16] Gutsche, Oliver, et al., "CMS Analysis and Data Reduction with Apache Spark," *Journal of Physics: Conference Series*, vol. 1085, no. 4, 2017.
- [17] Lakshmanan, Valliappa, *Data Science on the Google Cloud Platform*, s.l. : "O'Reilly Media, Inc.", 2017. 9781491974537.
- [18] "Google Cloud Documentation," retrieved 9 14, 2025, <https://cloud.google.com/docs>.

[19] Kumar, Ravi and Tripathi, Arun, *ROOT: A Data Analysis and Data Mining Tool from CERN*, 2008.

PRILOG

POPIS SKRAĆENICA

ATLAS

A Toroidal LHC ApparatuS - Toroidalni aparat LHC-a, 1, 5, 7, 9

BDT

Boosted Decision Tree - Pojačano stablo odlučivanja, 6, 7

c

Brzina svetlosti, 3, 24, 26, 30, 34

CERN

Organisation européenne pour la recherche nucléaire – Evropska organizacija za nuklearno istraživanje, A, 1, 2, 5, 6, 7, 9, 19, 24, 35, 39, 40, 41, 52, 53, 57

CLI

Command Line Interface - Komandna linija, 16

CMS

Compact Muon Solenoid - Kompaktni mionski solenoid, 5, 6, 7, 9, 19, 20, 21, 24, 27, 28, 29, 34, 35, 36, 40, 41

CMSSW

CMS Software - CMS softver, A, 1, 21, 22, 25, 27, 28, 29, 38, 41, 42, 43

EOS

Extensible Object Store, distribuirani sistem fajlova koji CERN koristi za skladištenje i pristup velikim datasetovima, 40, 41, 42, 43, 48

fb

Femtobarn, 20, 22, 34

GCP

Google Cloud Platform, 35

GeV

Gigaelektrovolt, 5, 19, 20, 21, 24, 26, 29, 30, 32, 34

GSL

GNU Scientific Library - GNU naučna biblioteka, 14

GUI

Graphical User Interface - Grafički korisnički interfejs, 17, 18

HDFS

Hadoop Distributed File System - Distribuirani fajl sistem Hadoop-a, 7

https

HyperText Transfer Protocol Secure - Bezbedni protokol za prenos hiperteksta, 15, 19, 22, 25, 26

itd.

i tako dalje, 29, 31

LHC

Large Hadron Collider - Veliki hadronski sudarač, 5, 6, 7, 9, 20

MC

Monte Carlo, 2, 24, 27, 28, 30, 34, 38, 40, 43, 45, 46, 47, 52

ML

Machine Learning - Mašinsko učenje, 7, 8

NAT

Network Address Translation - Prevođenje mrežnih adresa, 2, 43, 44

npr.

na primer, 10, 11, 13, 14, 15, 16, 22, 29, 31

PAW

Physics Analysis Workstation - Radna stanica za analizu podataka u fizici, 9

SITECONF

Site Configuration, struktura konfiguracionih fajlova u CMS okruženju koja definiše pristup podacima i lokalne parametre rada, 41

SPMD

Single Program, Multiple Data- Jedan program, više podataka, 36

SQL

Structured Query Language - Struktuirani jezik upita, 7

TeV

Teraelektrovolt, 20, 24, 34

tj.

to jest, 11, 35

tzv.
 takozvani, 6, 7, 9, 10, 12, 13, 23, 25, 28, 35, 38, 40
 VM
 Virtual Machine - Virtualno okruženje, 21, 37, 40, 41, 42, 43, 44
 WSL
 Windows Subsystem for Linux - Windows podsistem za Linux, 16, 17

POPIS SLIKA

Slika 1. Arhitektura ROOT okvira (izvor: [14])	10
Slika 2. Struktura drva (izvor: [14])	13
Slika 3. ROOT CLI	16
Slika 4. Demo prozor u ROOT-u	17
Slika 5. ROOT GUI	17
Slika 6. VirtualBox sa CMS VM image-om	21
Slika 7. Pokretanje CSM virtualne mašine	21
Slika 8. Skinuti ROOT fajlovi	22
Slika 9. Rezultati sa očekivanom brzinom obrade ($\sim 0,0087$ s po događaju)	49
Slika 10. Rezultati sa prosečnim vremenom obrade približno duplo u odnosu na očekivano vreme	49
Slika 11. Rezultati sa prosečnim vremenom obrade oko tri puta veće od planirane vrednosti	50
Slika 12. Prikaz troškova obrade na Google Cloud Platformi	50

POPIS GRAFIKA

Grafik 1. Primer histograma u ROOT okviru (izvor: https://root.cern/gallery/#data-analysis-and-visualization)	15
Grafik 2. Objavljena vizualizacija rezultata (izvor: [1])	19
Grafik 3. Rezultat sa podacima iz Monte Karlo simulacije	20
Grafik 4. Novi dijagram nakon trećeg nivoa	27
Grafik 5. Rezultat sa generisanim MC podacima	46
Grafik 6. Rezultat sa delimično uključenim stvarnim podacima	47
Grafik 7. Rezultat sa svim generisanim podacima	48

POPIS LISTINGA

Listing 1. Koraci na komandnoj liniji za drugi nivo replikacije	22
Listing 2. Primer obrade podataka sa parametrima	24
Listing 3. Koraci u komandnoj liniji za treći nivo replikacije	25
Listing 4. Podešen JSON validacioni fajl i izvor podataka	27
Listing 5. Podešen izvor podataka	27
Listing 6. Pozivanje fajla HiggsDemoAnalyzer	28
Listing 7. Struktura klase HiggsDemoAnalyzer	29
Listing 8. Konstruktor klase HiggsDemoAnalyzer	30
Listing 9. Inicijalizacija promenljive u analize metodi	31
Listing 10. Validacija i filtriranje podataka	31
Listing 11. Selekcija kandidata za elektron parove	33
Listing 12. Spajanje histograma	34
Listing 13. Podešavanje izvor podataka u demoanalyzer Python fajlovima	38
Listing 14. Skripta za dohvatanje index fajlova	39
Listing 15. Podešavanje okruženja i CMS Shell omotača u kontekst fajlu	40
Listing 16. Podešavanje mapiranja skladišta podataka u kontekst fajlu	41
Listing 17. Podešavanje ličnog startup skripta i CernVM konfiguracije u kontekst fajlu	42
Listing 18. Komande za podešavanje NAT gateway-a	43
Listing 19. Skripta za kreiranje virtualnih mašina	43
Listing 20. Podešavanje maxEvents i skipEvents parametara	44
Listing 21. Podešavanje ulaznih fajlova - generisani MC podaci	46
Listing 22. Podešavanje ulaznih fajlova - delimično uključeni stvarni podaci	47
Listing 23. Podešavanje ulaznih fajlova - svi generisani podaci	48

POPIS FORMULE

Formula 1. Formula za skaliranje podataka 23

RADNA BIOGRAFIJA (CV)

Ime i prezime: Viktor Varkulja

Datum i mesto rođenja: 20. januar 2002, Subotica, Srbija

Adresa: Srednja 50, 24000 Subotica, Srbija

Kontakt: viktor.varkulja@gmail.com | +381 606864452

OBRAZOVANJE I USAVRŠAVANJE

Master strukovni inženjer informacionih tehnologija ITS – Visoka škola informacionih tehnologija, Beograd (2023 – u toku), EOK nivo 7

Strukovni inženjer informacionih tehnologija ITS – Visoka škola informacionih tehnologija, Beograd (2020 – 2023), EOK nivo 6

RADNO ISKUSTVO

IT Generalist – Luxoft d.o.o., Beograd (avgust 2024 – januar 2025) Obuka i praktičan rad iz oblasti IT infrastrukture, DevOps-a, cloud computing-a i automatizacije procesa.

Associate – ITPRO d.o.o., Subotica (april 2021 – trenutno) Rad na integraciji podsistema za naplatu i plaćanje sa SEF i ePP sistemima, generisanje XML i JSON podataka iz baza.

QA Tester Intern – Comtrade System Integration, Beograd (maj 2023) Automatizovano testiranje veb-aplikacija pomoću Selenium-a i GitHub-a, sa fokusom na čist kod i timski rad.

PROJEKTI I PROFESIONALNE AKTIVNOSTI

Replikacija raspada Higsovog bozona u četiri leptona (2025) – Analiza CERN Open Data uz ROOT, C++ i Python.

Veb aplikacija za Gradski muzej Subotica (2024) – ASP.NET MVC i C#, sa CMS sistemom i administratorskim panelom.

Vinarija Varkulja (2024) – Laravel aplikacija za praćenje vinograda i online prodaju vina (AngularJS, MySQL).

Projekat baze podataka za Gradski muzej Subotica (2022) – Relaciona baza podataka, SQL upiti i procedure.

KONFERENCIJE I SEMINARI

CERN School of Computing (2025) – Softversko inženjerstvo i tehnologije podataka.

LINKIT&EdTech25 konferencija (2025) – Presentacija projekta replikacije Higsovog bozona.

CERN Thematic School of Computing (2024) – Naučni softver za heterogene arhitekture.

STRUČNE I TEHNIČKE VEŠTINE

Programski jezici: C#, C++, Python, Java, Go, SQL

Alati: Git, Selenium, REST API-jevi, Docker, MS Office, Google Suite

Platforme: Unix/Linux, ASP.NET, razvoj mobilnih aplikacija (osnovno), osnove AI i mašinskog učenja

JEZICI

Srpski – maternji

Mađarski – maternji

Engleski – C1 nivo (govor, pisanje, tehnička terminologija)