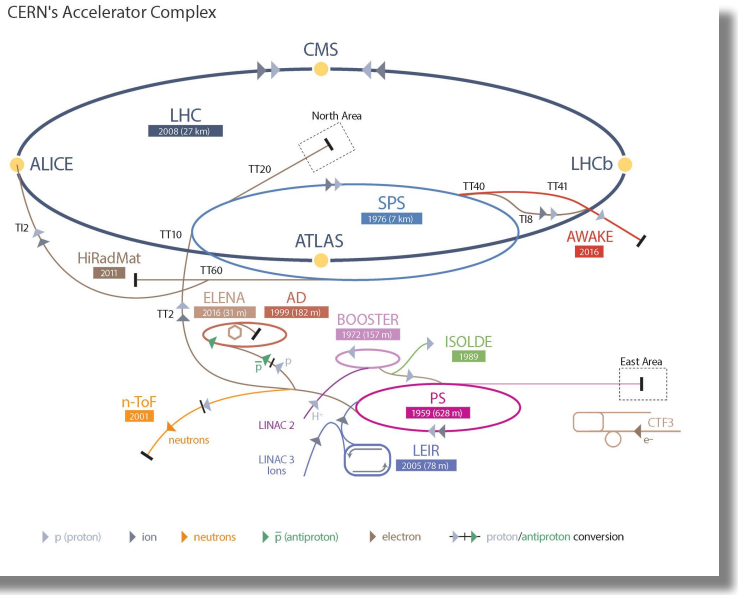
Adatelemzés és számítástechnika

Krasznahorkay Attila

Hungarian Teacher Programme

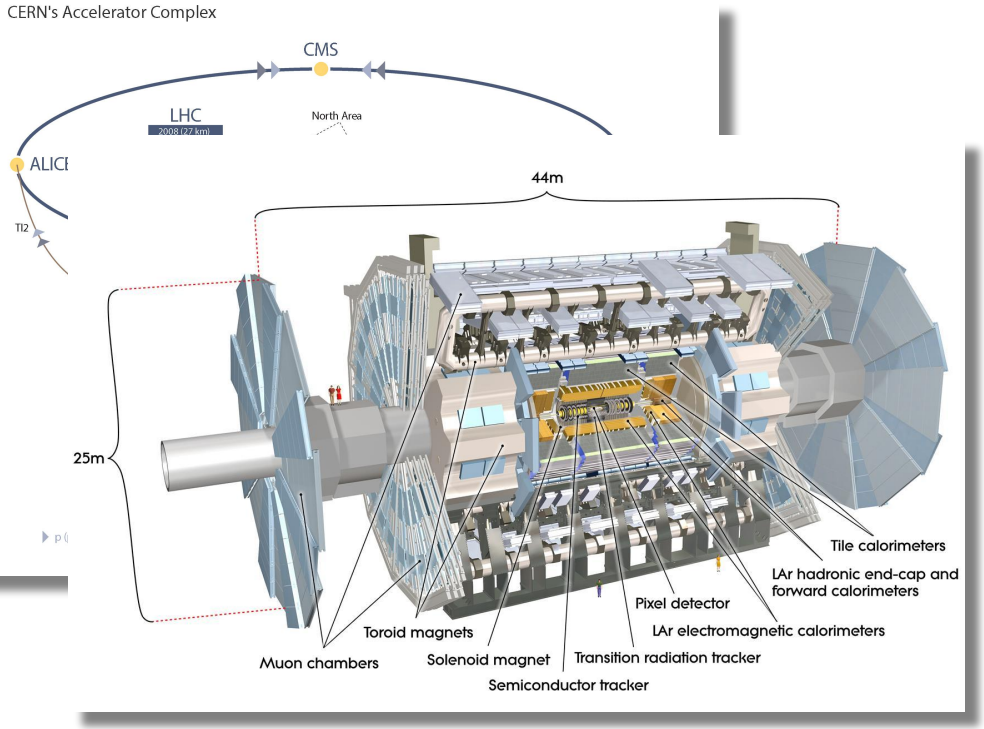
Emlékeztető

A madártávlat



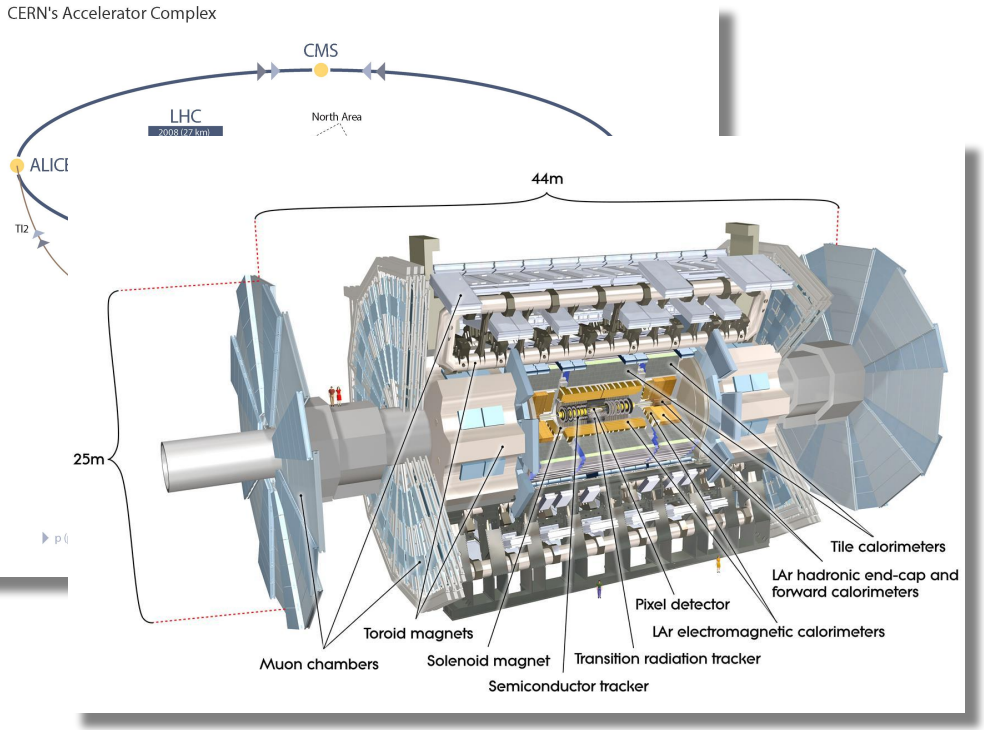
Létrehozzuk az “érdekes” reakciókat
(Barna Dániel előadása)

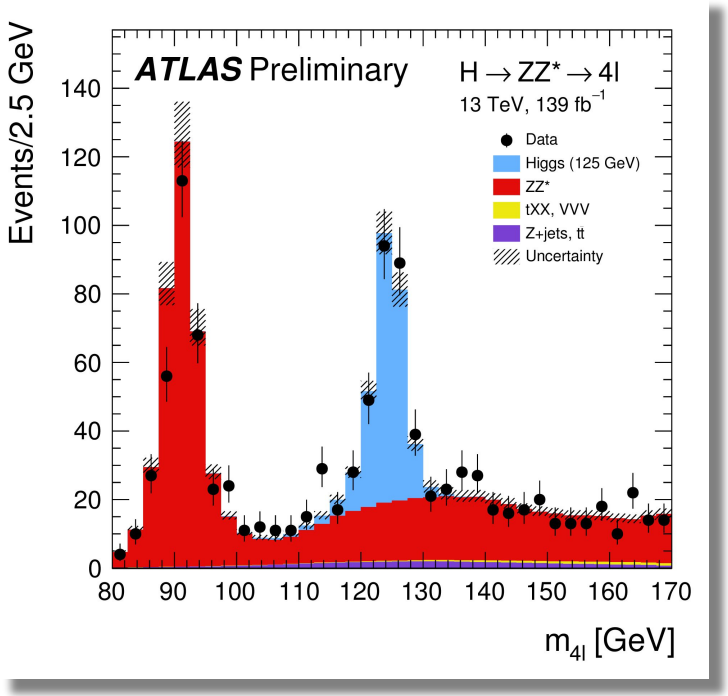
A madártávlat



Érzékeljük a kijövő részecskéket
(Barna Dániel előadása)

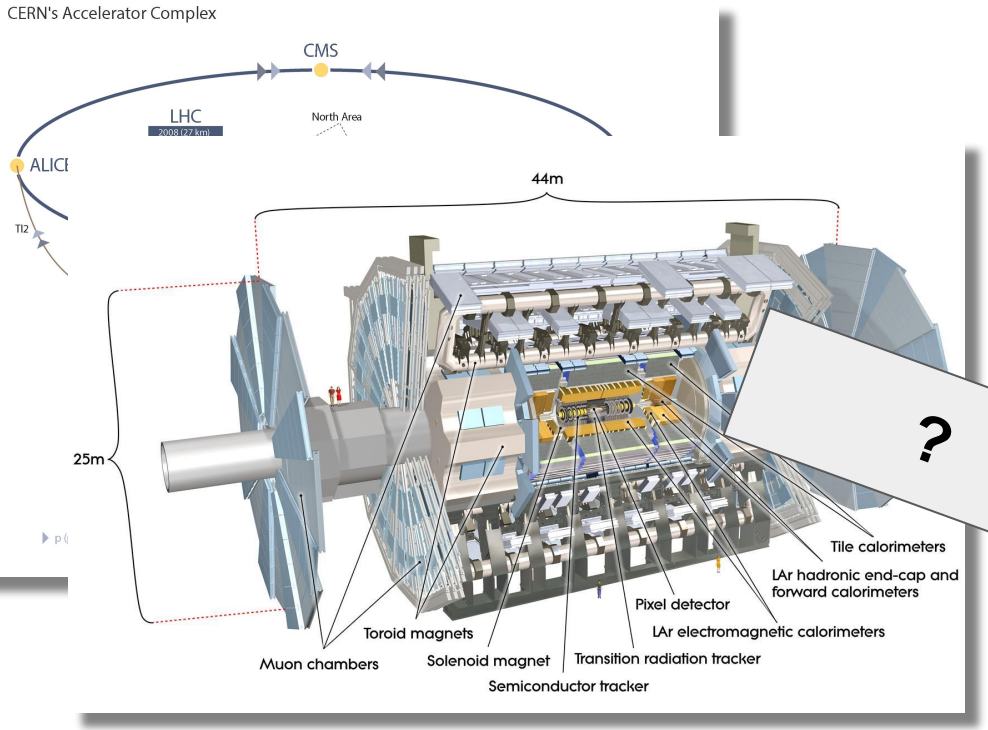
A madártávlat



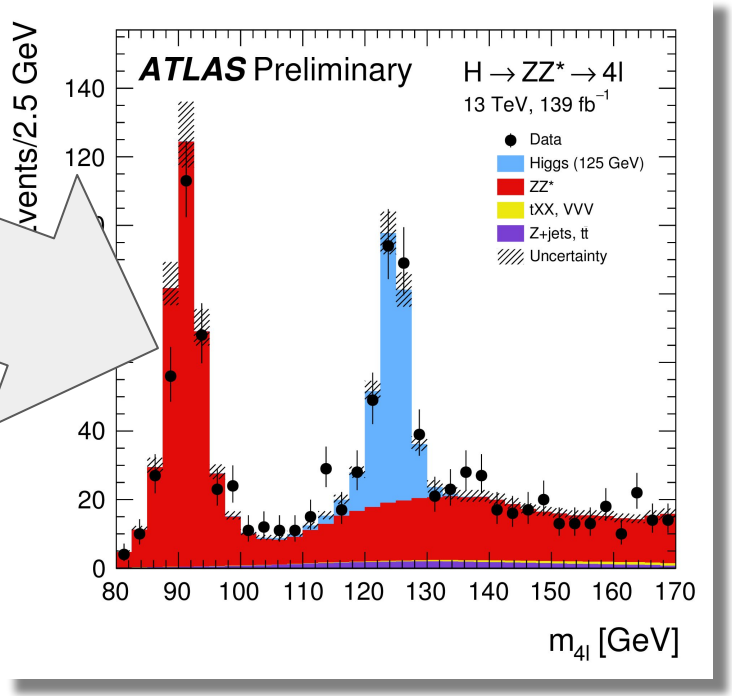


Levonjuk a fizikára vonatkozó következtetéseket
(Újvári Balázs előadásai)

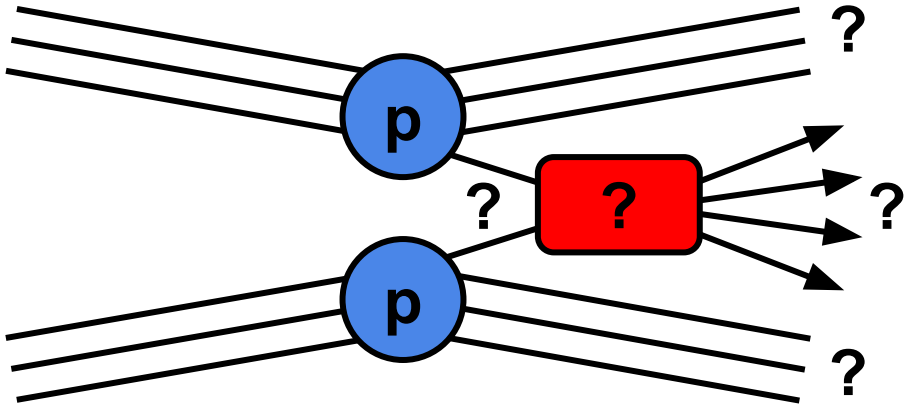
A madártávlat



?



A kísérlet



- Amire igazán kíváncsiak vagyunk az LHC-nál az a piros dobozban végbemenő folyamat
- Az alapján amit a reakcióból kijönni látunk, következtetünk vissza, hogy:
  - Milyen proton-proton ütközést figyeltünk meg
  - Milyen reakció zajlott le pontosan

Az adatgyűjtés

- A detektorokkal egymástól független “eseményeket” rögzítünk
  - Ahogyan Dani bemutatta, minden detektor típus rá jellemző módon tudja érzékelni azt ha különböző részecskék kölcsönhatásba lépnek vele
  - A rögzített információt, mint egy fényképezőgép CCD kameráját, kiolvassuk
- Az LHC másodpercenként >1 milliárd proton-proton ütközést hoz létre, amit mindet nem rögzíthetünk
  - Kb. 40 millió proton csomag ütközés jön létre másodpercenként, mindegyikben 40-60 proton-proton ütközéssel
  - Egy válogató (trigger) rendszerrel választjuk ki a fizika szempontjából legérdekesebb eseményeket, kb. 1000 darabot másodpercenként. Ez másodpercenként néhány GB adat gyűjtéséhez vezet.
- A “nyers” adatokat a CERN számítóközpontjába küldjük, ahonnan biztonsági másolatok készülnek róla a világ több pontjára
  - Évente több milliárd eseményt rögzítünk így

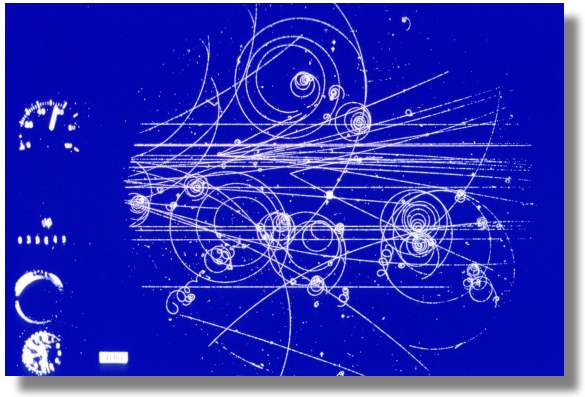
Az adatfeldolgozás

- A nyers adatokat erre fejlesztett célprogramokkal “rekonstruáljuk”. Így állapítva meg, hogy milyen részecskék milyen tulajdonságokkal repültek át (és a legtöbb esetben semmisültek meg) a detektoron/detektorban
  - Megkapjuk, hogy mennyi elektron, müon, hadronzápor, stb. keletkezett az egyes eseményekben, és milyen tulajdonságokkal
- Az egységesen rekonstruált adatokat több lépésen keresztül analizáljuk, az egyes analízisek kívánalmainak megfelelően
  - Például kereshetünk két leptonos eseményeket kevés “hiányzó energiával” ha Z bozonos eseményeket akarunk vizsgálni
  - A kiválasztott eseményekben egyenként kiszámíthatunk különböző komplex paramétereket (például az eseményben feltételezett Higgs bozon tömegét), amiknek az eloszlásából statisztikai analízissel a háttérben zajló fizikára tudunk visszakövetkeztetni.

Történelmi kontextus

Adat-analízis a múltban

- Érdemes látni, hogy hogyan is jutottunk idáig
  - A jelenlegi komplex adatfeldolgozás nem egyik percről a másikra lett kitalálva…
- Milyenek voltak a kísérletek a múltban?
  - **Legkorábban:** Fényképek és besugárzott minták szemmel tanulmányozása
    - Gáz és buborékkamra képeket ellenőriztek egyenként, betanított munkával
  - **Később:** Kevés paraméter (pl. energialeadás, pozíció) mérése, és közvetlen analizálása
    - Képesek voltak a különböző detektorokból kijövő elektromos jeleket olyan formára hozni, hogy azokból a detektor által érzékelt részecskék milyenségére tudjanak következtetni
    - De a jeleket lehetetlen volt egyesével rögzíteni, a kor számítástechnikája ezt nem engedte meg
    - Sokcsatornás analizátorokkal elektronikusan készítettek eloszlásokat a mért paraméterekből
  - **Végül:** Amint a mért elektronikus paramétereket egyenként rögzíteni lehetett, az adatok feldolgozása a kísérlet elvégzésétől későbbre tolódhatott
    - Lehetővé téve, hogy kifinomult adatfeldolgozással pontosabb méréseket lehessen elvégezni

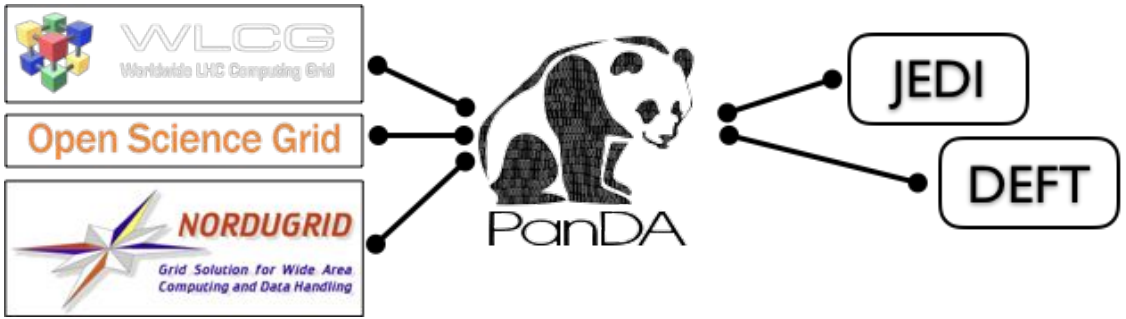


Számítástechnika a múltban

- A CERN első “valódi” számítógépe 1958-ban állt üzembe
  - Programokat és adatokat lyukszalagon tudott kapni, és természetesen mai szemmel nagyon lassú volt
- Az évek folyamán egyre nagyobb és nagyobb teljesítményű szinguláris gépeket helyeztek üzembe
  - https://cerncourier.com/a/computing-at-cern-the-mainframe-era
- A nagy változást a személyi számítógépek megjelenése hozta
  - Ezeket használni lehetett az adatgyűjtés irányításától egészen az adatfeldolgozás végső lépéséig
  - Azóta is a személyi számítógépek uralják a fizika kísérleteket
    - Bár a “szuperszámítógépek” mostanra újra népszerűek lettek

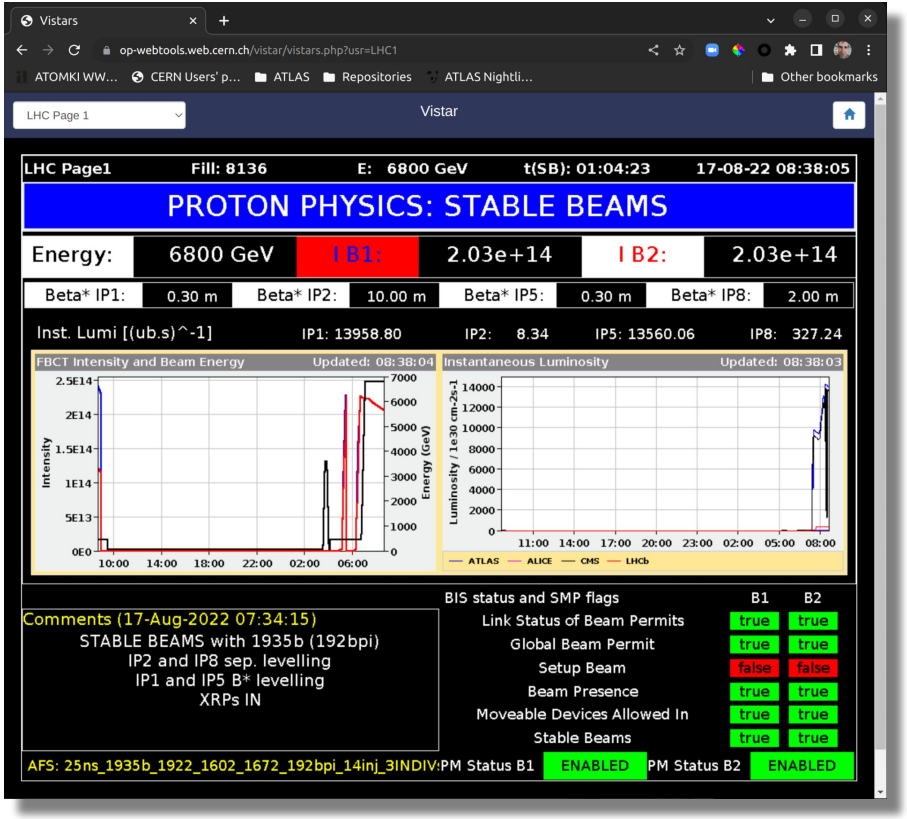
A grid létrejötte

- A személyi számítógépek előretörésével hamar megjelentek az ezekből épített klaszterek
  - Ezek arra lettek kitalálva, hogy ugyanazt a feladatot, egyenként kis változtatásokkal, sok gépen/processzoron tudják egyszerre futtatni
  - A LEP adatait is ilyen klasztereken dolgozták fel annak idején
- Az LHC tervezésénél hamar világossá vált, hogy nem célszerű egy nagy klaszterre bízni az adatfeldolgozást → A világ sok egyéni klaszterét kell inkább összekötni egy egységes rendszerré
  - Amire úgy lehet feladatokat küldeni, hogy nem kell tudjuk az adatok hol vannak, és a programunk pontosan hol fog lefutni

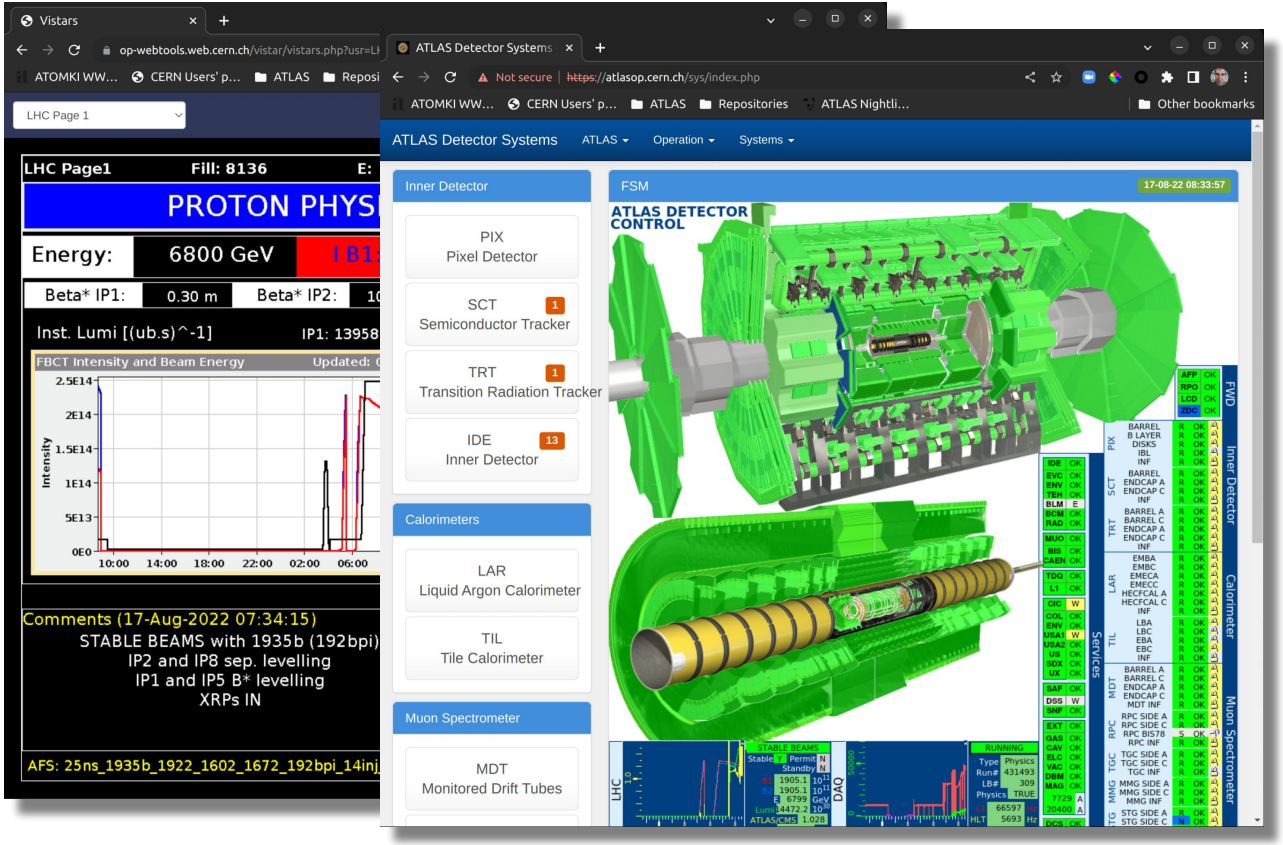


Modern adatfeldolgozás

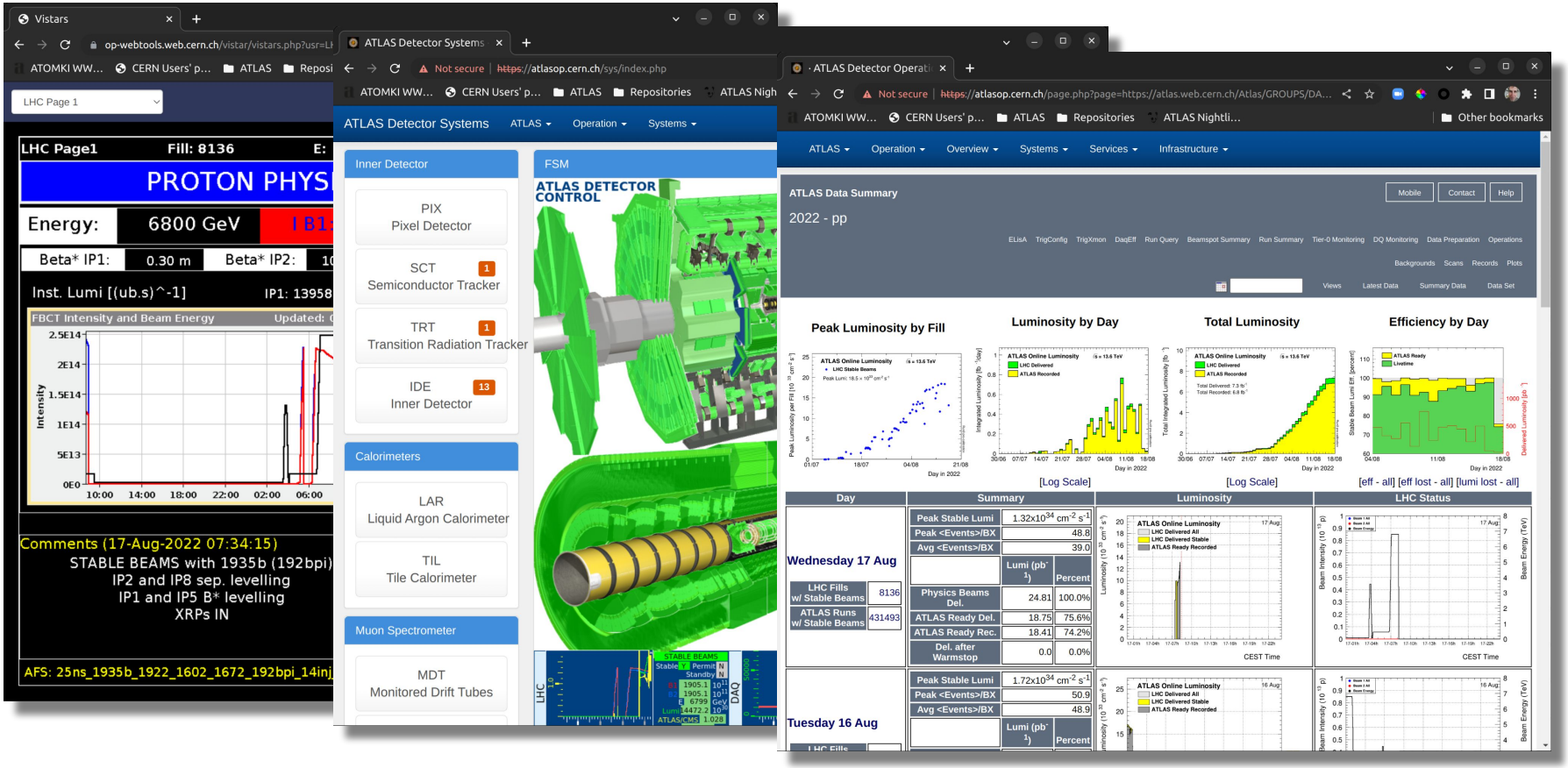
Az ATLAS adatgyűjtése



Az ATLAS adatgyűjtése

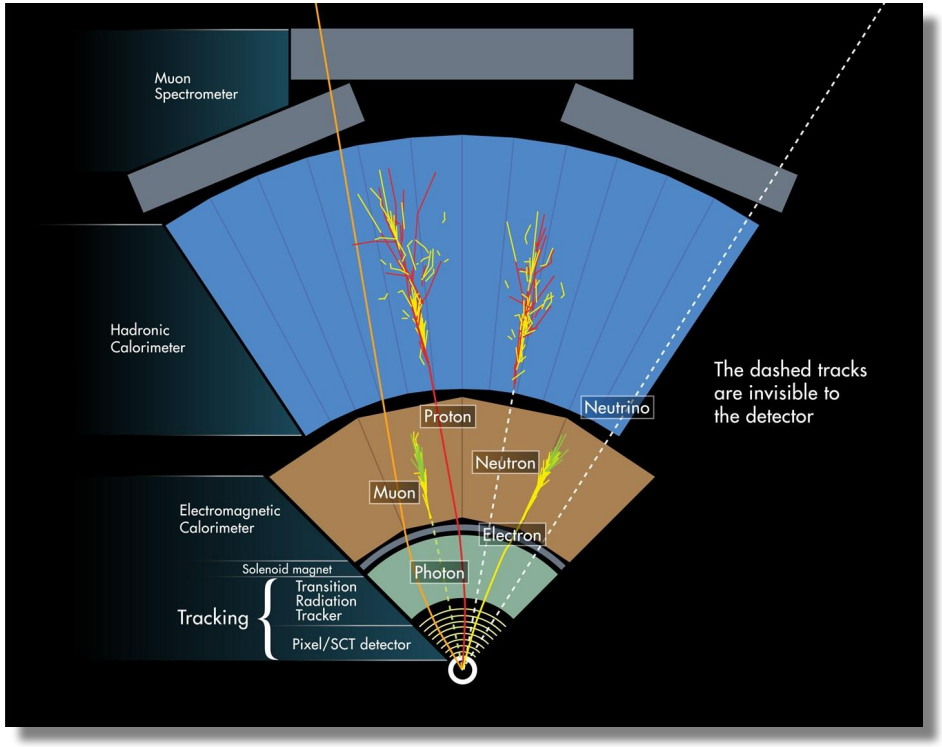


Az ATLAS adatgyűjtése



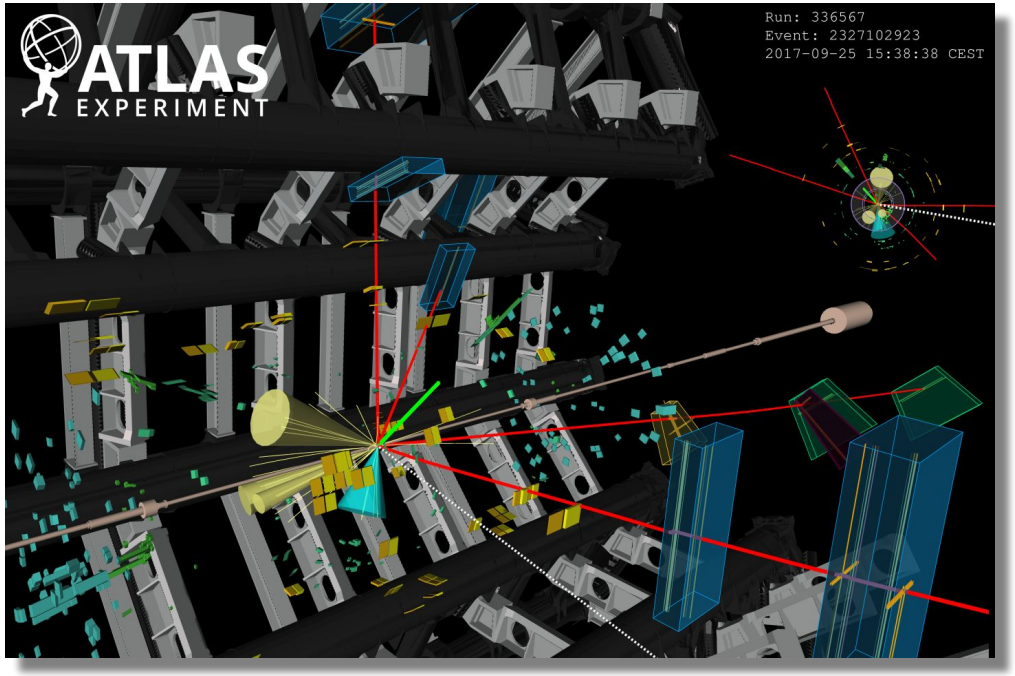
Rekonstrukció

- A detektorokat úgy építjük meg, hogy a nekünk érdekes részecskék egymástól különböző jeleket hagyjanak bennük
- Pl. egy elektron rekonstrukciójához:
  - Keresnünk kell egy töltött részecske nyomát a nyomkövető detektorban
  - Ehhez hozzá kell tudnunk rendelni egy energia-klasztert az elektromágneses kaloriméterben
    - A klaszter és töltött nyom tulajdonságaira sok technikai feltételt is szabunk
  - Nem szabad “aktivitást” találnunk a hadron kaloriméterben az elektron “mögött”



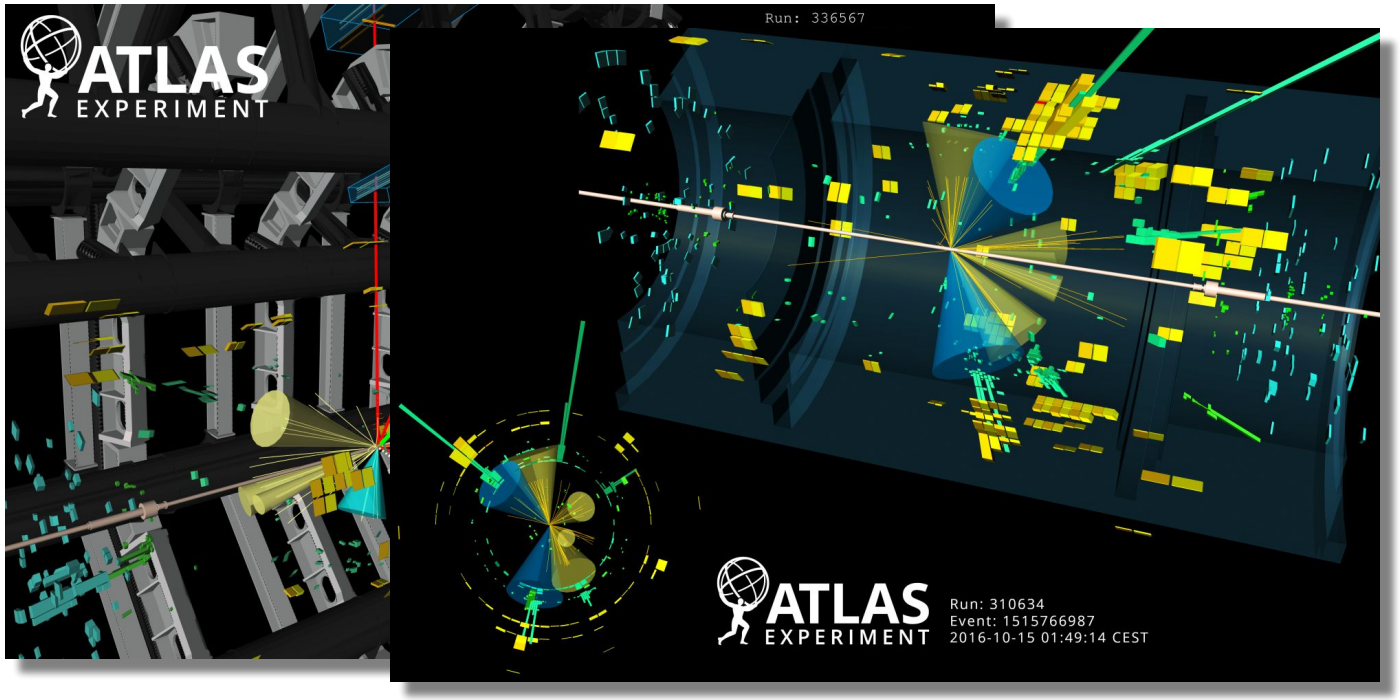
Trigger

- Ez történik legelőször, de egyszerűbb a rekonstrukció után megemlíteni
- Az érdekes eseményeket azok gyors rekonstrukciójával válogatjuk ki



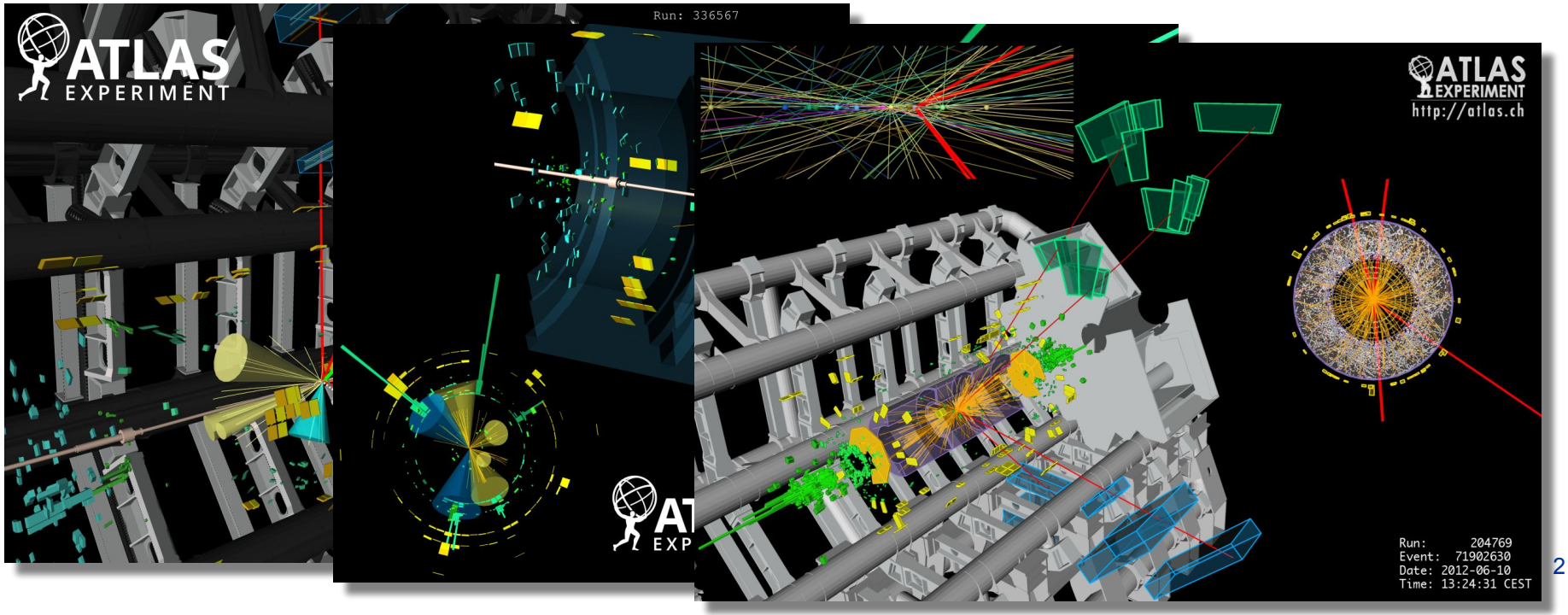
Trigger

- Ez történik legelőször, de egyszerűbb a rekonstrukció után megemlíteni
- Az érdekes eseményeket azok gyors rekonstrukciójával válogatjuk ki



Trigger

- Ez történik legelőször, de egyszerűbb a rekonstrukció után megemlíteni
- Az érdekes eseményeket azok gyors rekonstrukciójával válogatjuk ki

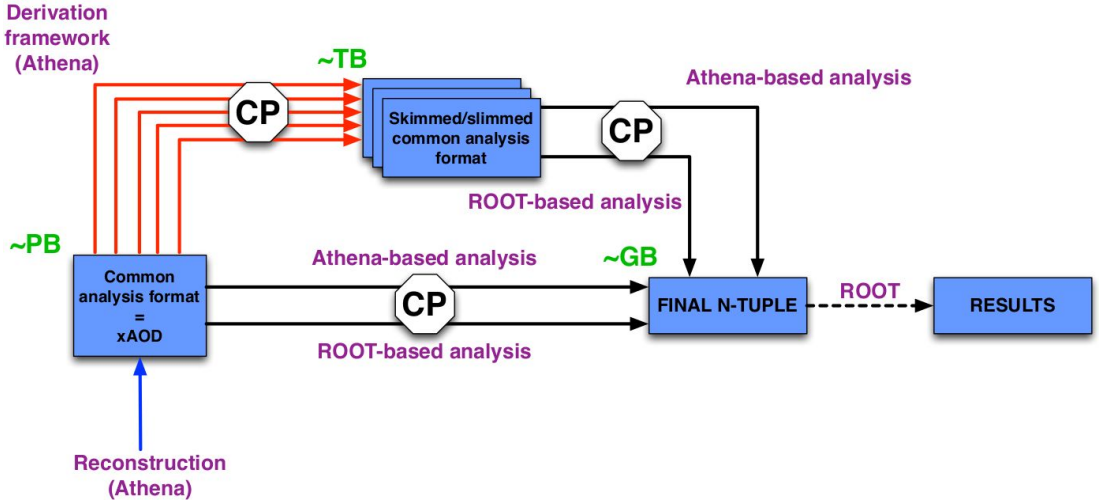


Szimuláció

- A rögzített adatokat a legtöbb esetben csak szimulációk segítségével tudjuk értelmezni
  - Sokszor a legegyszerűbb fizikai folyamatok is annyira bonyolult módon hatnak kölcsön a detektorokkal, hogy csak így tudjuk megbecsülni, mit is mértünk pontosan
- Sok lépésből áll. Az érdekesek ezekből:
  - **Esemény generálás:** Elméleti fizikus kollégák írnak rá szoftvert, hogy véletlenszerű statisztikával valamilyen fizikai folyamathoz “eseményeket” gyártsunk.
  - **Detektor szimuláció:** A generált események “stabil” részecskéinek a kölcsönhatását modellezzük a teljes detektorral. Szimulálva a detektor minden hibáját, és az LHC beállításait is.
    - Itt vesszük figyelembe, hogy mostanra >60 proton-proton ütközés is történhet egyszerre egy-egy eseményben
  - **Trigger és rekonstrukció:** A szoftverünket ugyanúgy futtatjuk a szimuláció kimenetén mint a valódi, a detektorból gyűjtött adatokon

Analízis modell

- Nem engedhetjük meg, hogy minden fizikus O(100) PB adatot dolgozzon fel minden alkalommal amikor változtat valamit az analízisén
  - Ehhez a Föld összes számítási kapacitása sem lenne elég
- Az analízis “elejét” szervezett formában végezzük a legtöbb kísérletben
  - Az eredetinél sokkal kisebb adatmennyiséget adva az egyes analizátoroknak, amiket már néhány nap alatt feldolgozhatnak a grid-en

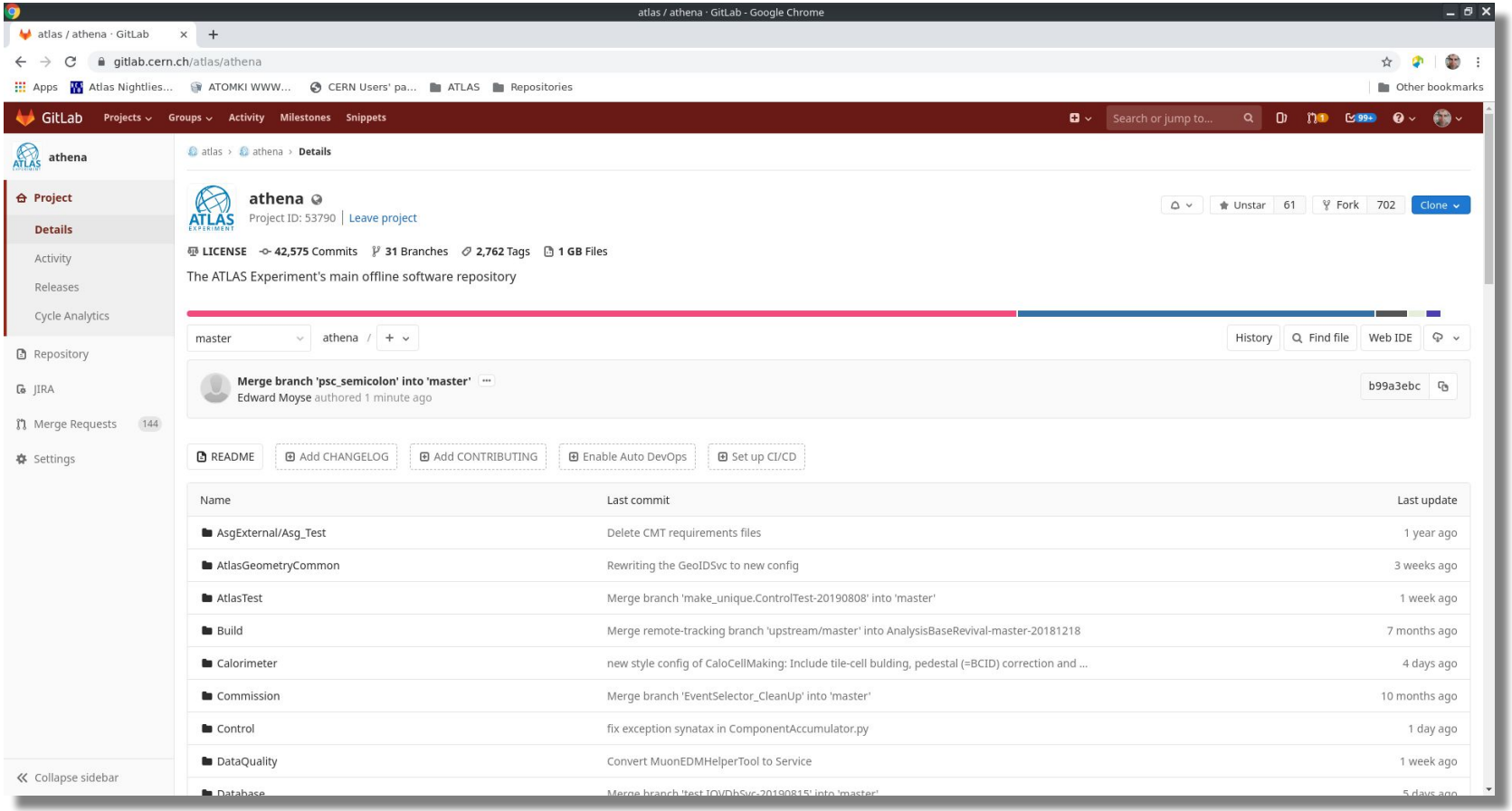


Mit csinál a fizikus?

- Kiválasztja, hogy milyen adatszettet fog feldolgozni
  - Egyúttal azt is, hogy milyen triggert fog használni, ami hatékonyan őrizte meg a számára érdekes fizikai folyamat eseményeit
- Kiválasztja, hogy milyen szimulációkat kell figyelembe vennie az analízisében
  - Ez magában foglalja az analízis által keresett fizikai folyamat szimulációját, és minden más “háttér” folyamatot amire érzékeny lehet az adatfeldolgozás
- Lefuttatja az analízis szoftverét ugyanolyan beállításokkal minden adatra és szimulációra, (relatíve) apró adatfájlokat létrehozva
  - Sokszor ez több lépésben történik, de végül különböző mennyiségek eloszlásait határozzák meg az analízisek a legtöbb esetben
- “Statisztikai analízist” végez a kapott eloszlásokon
  - Megmérendő valamilyen paramétert, vagy felfedezési/kizárási valószínűséget meghatározva
- Elvégzi az egészet még N alkalommal, az analízis minden szisztematikus bizonytalanságát figyelembe véve

Modern számítástechnika

Az ATLAS offline szoftvere



Az ATLAS “offline” szoftvere

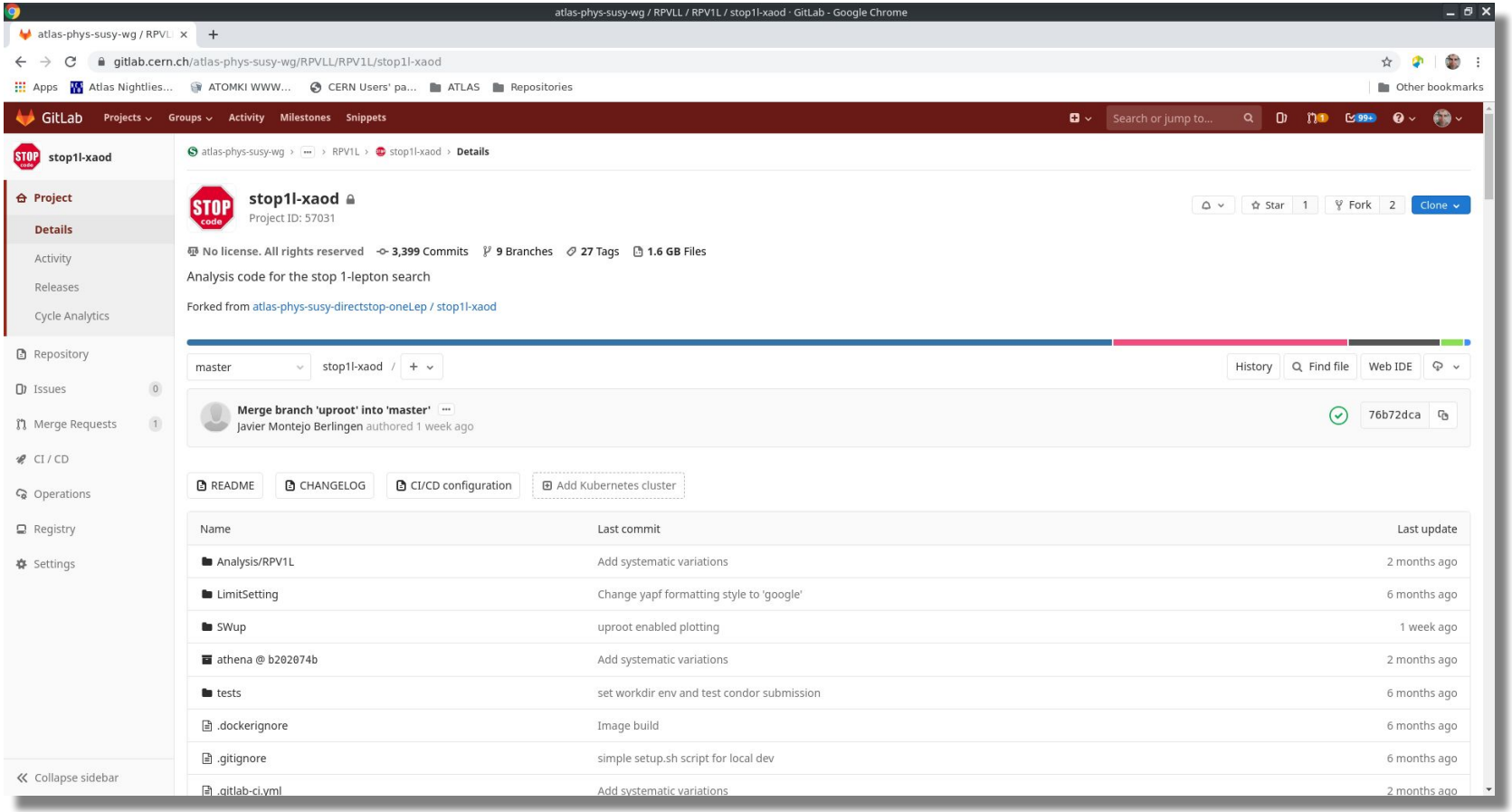
```
Totals grouped by language (dominant language first):
cpp:         3919711 (69.71%)
python:      1178662 (20.96%)
xml:          341767 (6.08%)
sh:            64815 (1.15%)
fortran:       58333 (1.04%)
ansic:         29040 (0.52%)
f90:           12340 (0.22%)
javascript:     6512 (0.12%)
php:            5487 (0.10%)
perl:           3571 (0.06%)
csh:             759 (0.01%)
pascal:          674 (0.01%)
java:            493 (0.01%)
awk:             343 (0.01%)
vhdl:            103 (0.00%)



Total Physical Source Lines of Code (SLOC)                = 5,622,610
Development Effort Estimate, Person-Years (Person-Months) = 1,731.67 (20,780.03)
 (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                         = 9.11 (109.31)
 (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 190.11
Total Estimated Cost to Develop                           = $ 233,924,916
 (average salary = $56,286/year, overhead = 2.40).
SLOCCount, Copyright (C) 2001-2004 David A. Wheeler
SLOCCount is Open Source Software/Free Software, licensed under the GNU GPL.
SLOCCount comes with ABSOLUTELY NO WARRANTY, and you are welcome to
redistribute it under certain conditions as specified by the GNU GPL license;
see the documentation for details.
Please credit this data as "generated using David A. Wheeler's 'SLOCCount'."
```

- Kb. 4 millió sor C++ és 1 millió sor Python
  - Óvatos becsléssel is >1000 emberi munkaév van benne
- Folyamatosan fejlesztjük, nekem is ez a fő feladatom az ATLAS-ban
- A teljes szoftvert több mint 5 óra megépíteni egy 16 processzoros gépen
  - Így a szoftverfejlesztést is egyedi módon kell végeznünk…
- Egy világméretű elosztott fájlrendszeren tároljuk
  - `/cvmfs/atlas.cern.ch`

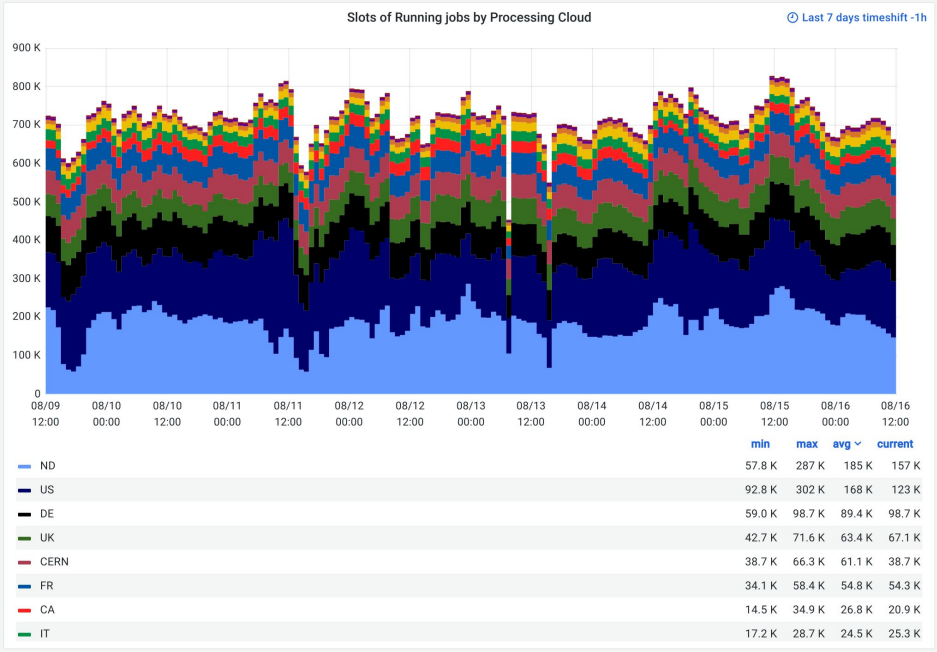
Analízis program-kód



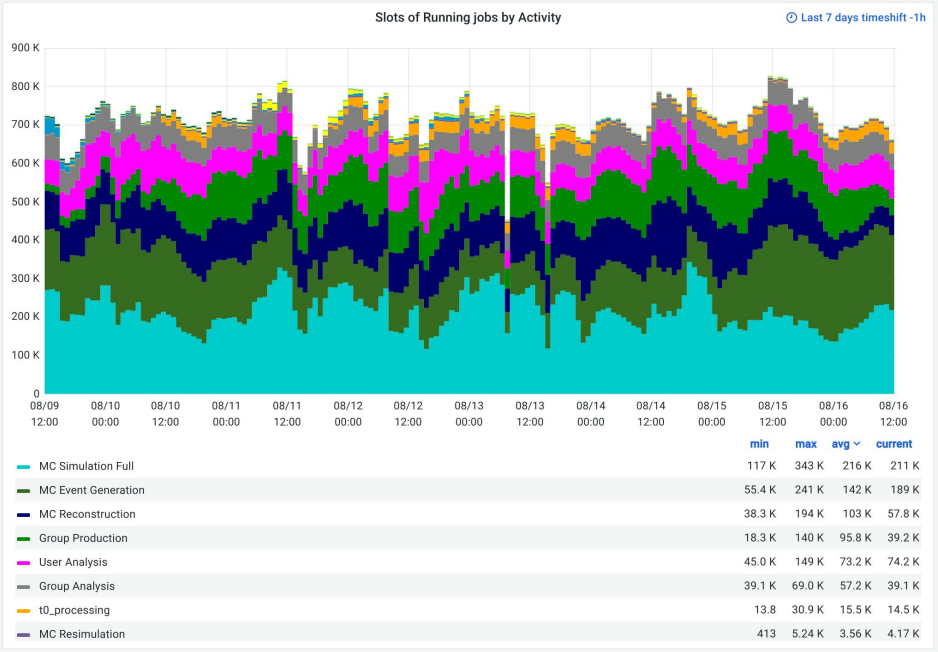
Gépi tanulás

- Nagyon kevés fizika analízis létezik manapság nélküle…
- Részecskefizikában 2 dologra használjuk a legtöbb esetben
  - **Klasszifikáció:** Egy-egy esemény tulajdonságai alapján azt döntjük el, hogy az olyan-e mint amit keresünk
    - A rekonstruált részecskék fajtáját is sokszor így határozzuk meg
  - **Regresszió:** Részecskék / események tulajdonságait számoljuk ki “alacsonyabb rendű” tulajdonságai alapján. Pl. meghatározzuk egy hadronzápor pontos energiáját a kaloriméterekben érzékelt jelekből.
- Ilyen technikákat már nagyon régóta használunk a fizikában.
  - A múltban ez főleg döntési fákat és neurális hálókat jelentett
- Az utóbbi 5-10 évben nagy fejlődésen ment keresztül
  - Mind elméleti, mind gyakorlati szempontból
  - Mostanra “az ipar” által készített kódokat szeretjük a leginkább használni
    - A Google által fényképek azonosítására használt kód nagyon jól használható az LHC eseményeinek megértésére is 😛

Mekkora is az LHC/ATLAS grid?



	min	max	avg	current
ND	57.8 K	287 K	185 K	157 K
US	92.8 K	302 K	168 K	123 K
DE	59.0 K	98.7 K	89.4 K	98.7 K
UK	42.7 K	71.6 K	63.4 K	67.1 K
CERN	38.7 K	66.3 K	61.1 K	38.7 K
FR	34.1 K	58.4 K	54.8 K	54.3 K
CA	14.5 K	34.9 K	26.8 K	20.9 K
IT	17.2 K	28.7 K	24.5 K	25.3 K



	min	max	avg	current
MC Simulation Full	117 K	343 K	216 K	211 K
MC Event Generation	55.4 K	241 K	142 K	189 K
MC Reconstruction	38.3 K	194 K	103 K	57.8 K
Group Production	18.3 K	140 K	95.8 K	39.2 K
User Analysis	45.0 K	149 K	73.2 K	74.2 K
Group Analysis	39.1 K	69.0 K	57.2 K	39.1 K
t0_processing	13.8	30.9 K	15.5 K	14.5 K
MC Resimulation	413	5.24 K	3.56 K	4.17 K

Összefoglalás

- Mostanra a szoftverfejlesztés egy nagyon lényeges elemévé vált a fizikának
  - Gyakorlatilag semmilyen kísérlet nem létezhet saját adatgyűjtő és feldolgozó szoftver nélkül manapság
  - Az adatok feldolgozását speciális programok írásával végezzük
    - Minden LHC fizikusnak muszály legalább “elfogadhatóan” programozni tudnia
- Az LHC több száz petabájt adatát a világméretű grid sok-százezer processzorán dolgozzuk fel

http://home.cern