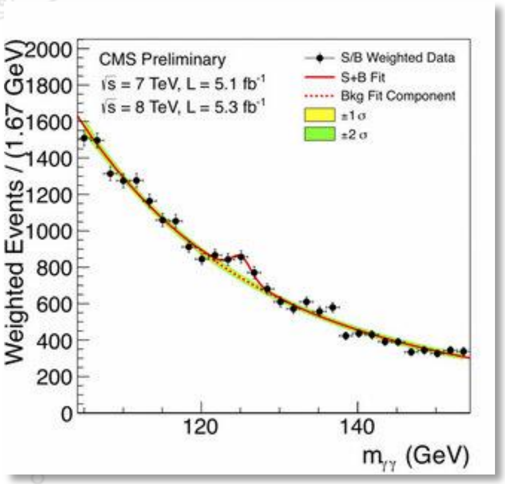
In-Database Physics Analysis

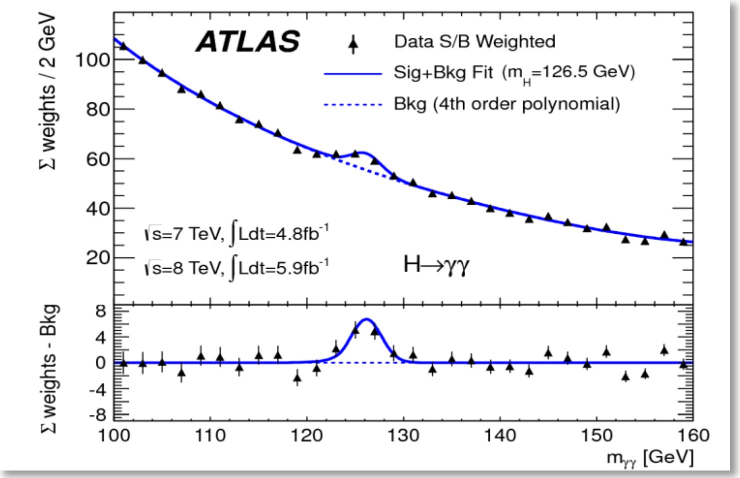
Maaike Limper

CERNopenlab

Higgs boson discovery

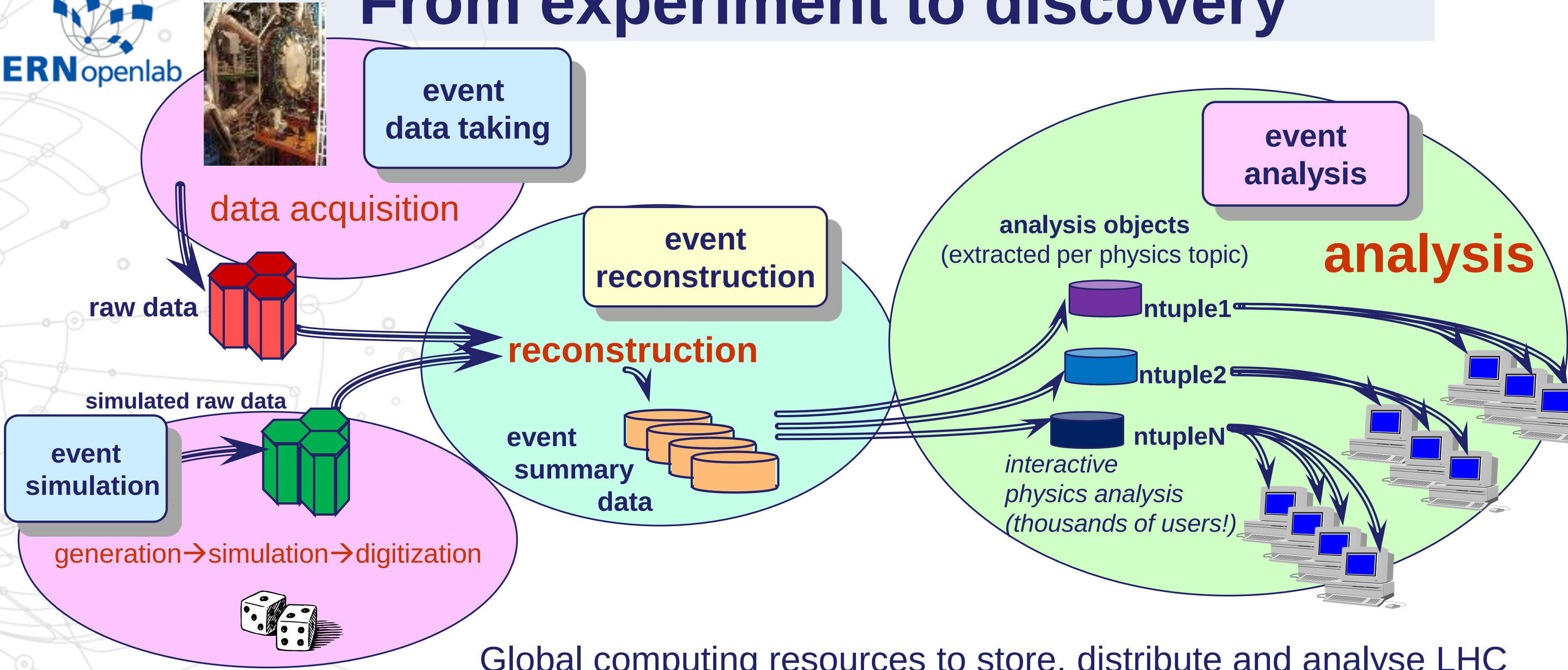
Plots of the invariant mass of photon-pairs produced at the LHC show a significant bump around 125 GeV …





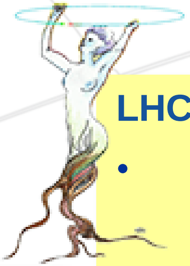
- 4 July 2012: The discovery of a "Higgs boson-like" particle!
- Operations of LHC and its experiments rely on databases for storing conditions data, log files etcetera
- … but the data-points in these plots did not came out of a database

From experiment to discovery



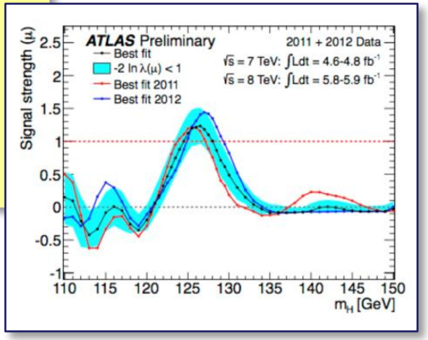
Global computing resources to store, distribute and analyse LHC data are provided by the Worldwide LHC Computing Grid (**WLCG**)

Data analysis in practice



LHC Physics Analysis is done with ROOT

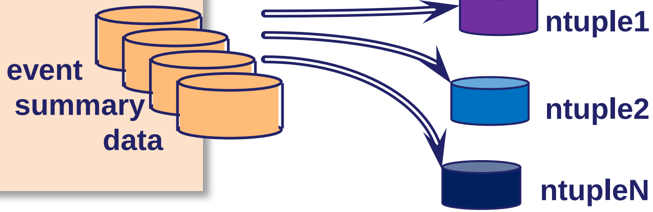
- Dedicated C++ framework developed by the High Energy Physics community, http://root.cern.ch
- Provides tools for plotting/fitting/statistic analysis etc.



ROOT-ntuples are centrally produced by physics groups from previously reconstructed event summary data

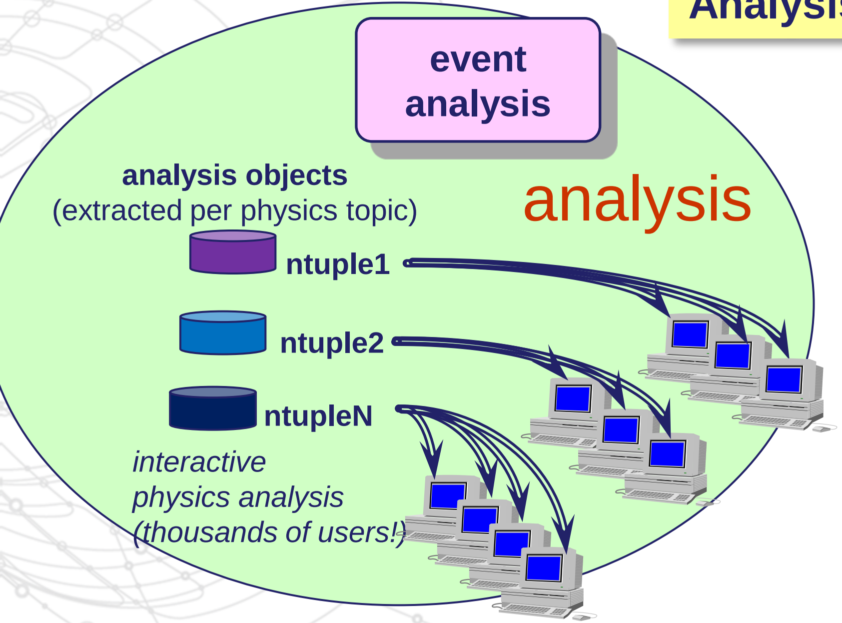
Each physics group determines specific content of ntuple

- *Physics objects to include*
- *Level of detail to be stored per physics object*
- *Event filter and/or pre-analysis steps*



Data analysis in practice

Analysis is typically I/O intensive and runs on many files



Small datasets→ copy data and run analysis locally

Large datasets:→use the LHC Computing Grid

- **Grid computing tools split the analysis job in multiple jobs each running on a subset of the data**
- Each sub-job is sent to Grid site where input files are available
- Results produced by sub-jobs are summed

…or

- ***Filter data and produce private mini-ntuples***
- ***Use local computer clusters***

My Openlab Project: ***Can we replace file-based analysis with a model where data is analysed inside a centrally accessible database?***

My test sample

- **Test sample of ATLAS analysis ntuples with 2012 collision data**
  - 127 ntuples (~200 GB) (‘NTUP_TOPMU’, ‘NTUP_TOPEL’)
  - .5% of entire dataset
- **These ntuples contain 4000 “branches” holding objects per events**
  - Objects can be float, int, double, etc but also vector or vector-of-vector of float, int, double, etc

Physicists don’t use all variables, they pick&choose to find variables giving best result for their analysis

ROOT-ntuple is designed to reduce I/O by loading only relevant branches

Test data stored in RDBMS

Store separate physics-objects in separate tables

- Allows users to only access objects relevant for their analysis
- Avoid storing vectors in columns to ensure easy predicate filtering

Table name	*columns*	*M rows*	*size in GB*
photon	216	89.9	**114.4**
electron	340	49.5	**94.6**
jet	171	26.8	**26.3**
muon	251	7.7	**14.2**
primary_vertex	25	89.5	**11.9**
EF (trigger)	490	7.2	**7.9**
MET_RefFinal	62	6.6	**2.3**
eventData	52	7.2	**1.4**

Physics Analysis C++

```
vector<float> el_pt;
vector<float> el eta;
tree->getBranch(“el_pt”,&el_pt);
tree->getBranch(“el_eta”,&el_eta);
//etc.
for ( ievent = 0 ; ievent<nevents ; ievent++){
    //find good electrons
    tree->NextEvent();
    for(i=0; i<nelectrons; i++){
      if( el_pt[i] > 25.  && fabs(el_eta[i])<2.5 etc.)
            ngoodelectron++;
   }
```

SQL analysis

select “E”, “px”, “py”, “pz” from “electron” where “pt”>25. and abs(“eta”)<2.5 ...

In-database physics analysis

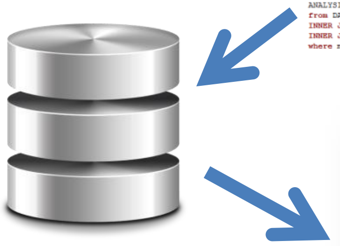
Physics Analysis database

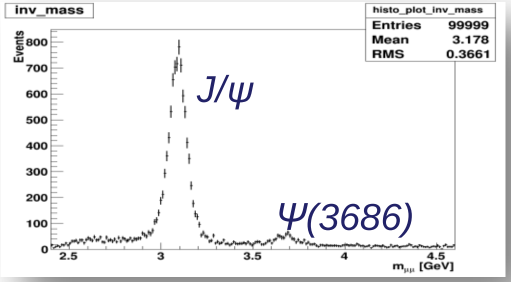
- Separate physics-objects in separate tables
- Physics-object described by <u>hundreds of variables</u> →wide tables!

Analysis queries

- Predicate filtering to quickly apply object quality-criteria
- Each analysis-specific query uses unique combination of columns

```sql
with
"sel_muon" as (select "muon_i","RunNumber","EventNumber","E","px","py","pz","charge","pt","phi","eta" from DATA12_8TEV."muon"
where "pt" > 10000. and abs("eta") < 2.7 and "tight" = 1  and ("id_d0"<10. or abs("eta")>2.5) and "ptcone20"<0.1*"pt")
select "RunNumber","EventNumber",mu_sel_n,
       muon0."muon_i" as mu_id0, muon1."muon_i" as mu_id1,
       muon0."charge" as mu_charge0, muon1."charge" as mu_charge1,
       muon0."pt"/1000. as mu_pt0, muon1."pt"/1000. as mu_pt1,
       muon0."eta" as mu_eta0, muon1."eta" as mu_eta1,
(case when abs(muon0."phi"-muon1."phi")<acos(-1.) then sqrt(POWER(abs(muon0."phi"-muon1."phi"),2)+POWER(abs(muon0."eta"-muon1."eta"),2))
        else sqrt(POWER( 2.*acos(-1.) - abs(muon0."phi"-muon1."phi"),2)+POWER(abs(muon0."eta"-muon1."eta"),2)) end) as DELTAR,
ANALYSISTOOLS.PHYSANALYSIS.INV_MASS_LEPTONS(muon0."E",muon1."E",muon0."px",muon1."px",muon0."py",muon1."py",muon0."pz",muon1."pz")/1000. as INV_MASS
from DATA12_8TEV."periodAllYear_v47-pro13-01"
INNER JOIN (select "RunNumber","EventNumber",COUNT(*) as mu_sel_n from "sel_muon" group by ("RunNumber","EventNumber")) USING ("RunNumber","EventNumber")
INNER JOIN "sel_muon" muon0 USING ("RunNumber","EventNumber") INNER JOIN "sel_muon" muon1 USING ("RunNumber","EventNumber")
where muon0."muon_i"<muon1."muon_i" and muon0."charge" != muon1."charge" and mu_sel_n=2;
```

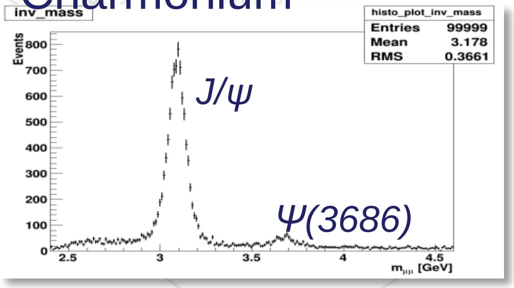




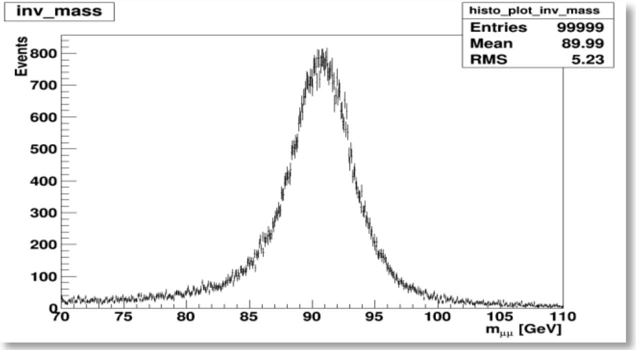
SQL analysis demo

- **Demonstrating how to produce some basic analysis plots with SQL**

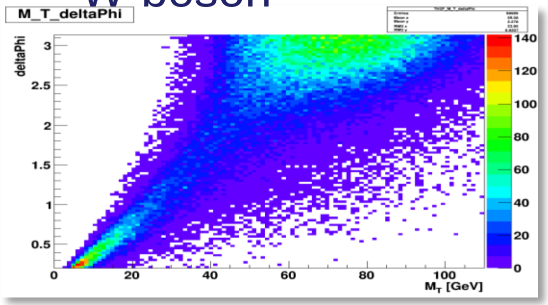
Charmonium



Z-boson

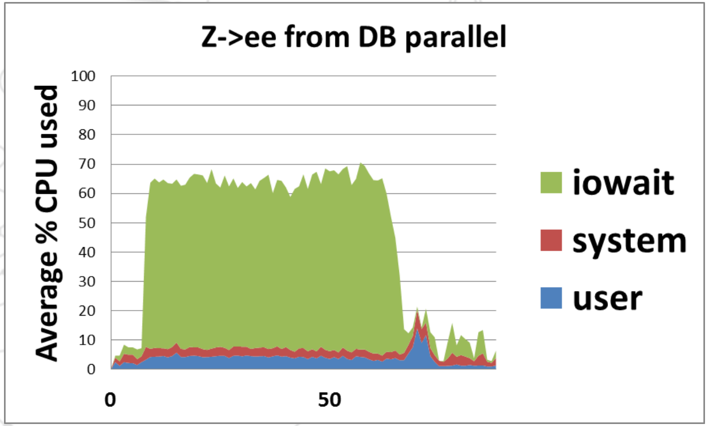


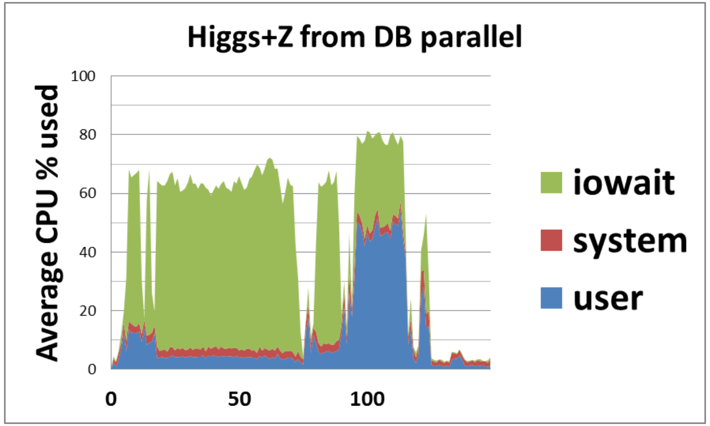
W-boson



The problem

- **Row-based storage means performance limited by I/O reads**
  - Full table scans over tables with many columns, while only few columns are used for each specific analysis
- **Combination of columns unique for each query**
  - Can't index every column!





Column vs row storage

- **Column storage stores column data together to reduce number of reads when few columns are needed**

select "E", "pt", "eta" from "particle"

row storage reads:

"E"	"pt"	"phi"	"eta"	"charge"	"author"	"ptcone20"	"ptcone30"
8163.1	8116.7	-0.882	0.107	1	3	6526.4	7823.5
8196.1	8046.1	2.18	0.193	1	3	0	0
4221.3	4172.5	-0.908	0.153	-1	1	0	0
72320.2	33146.4	-0.829	-1.416	1	3	0	0
6236.5	2759.1	1.169	1.456	-1	2	0	0
205693.7	16607.2	1.904	3.208	0	8	-999	-999
395287.4	13725.6	1.486	4.053	0	8	-999	-999
4506	3520.2	0.328	-0.732	1	1	26672.3	29752.8
258925	10522.7	1.213	-3.896	0	8	-999	-999

column storage reads:

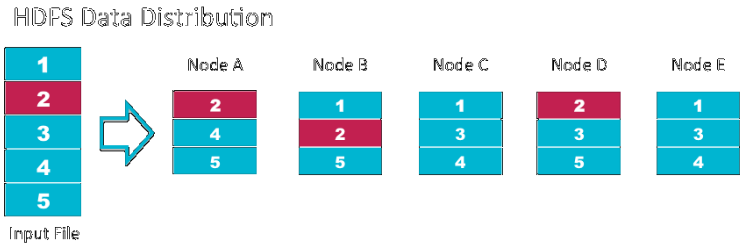
"E"	"pt"	"eta"
8163.1	8116.7	0.107
8196.1	8046.1	0.193
4221.3	4172.5	0.153
72320.2	33146.4	-1.416
6236.5	2759.1	1.456
205693.7	16607.2	3.208
395287.4	13725.6	4.053
4506	3520.2	-0.732
258925	10522.7	-3.896

Other databases

- **Analysing Big Data sets is a real-world problem**
- **In recent years many new (non-relational) databases became available, such as Hadoop**
- **Tests on-going to combine SQL-approach using column storage**
  - Hadoop+Impala with Parquet storage
  - Scalable Postgres DB with column store extension (CitusDB)

Other databases

- ***Hadoop+Impala***
  - *Hadoop File Storage System (HDFS) divides input in blocks, divided over nodes*
  - *Cloudera Impala: query-engine on Hadoop*
  - *Using Parquet-format (column-storage) to store data in HDFS*



- ***Scalable Postgres (CitusDB)***
  - *Workers on nodes represent independent Postgres instances*
  - *Master-node chop input-table in sub-table divided over workers*
  - *Use CitusDB column-store extension for storing data*

Other databases

- ***Test setup with 4 nodes running simultaneously***
  - *Oracle 4-node RAC*
  - *Hadoop 4-node cluster*
  - *CitusDB 4-node cluster*

- ***DEMO: simple query performance comparison using different database systems***

LHC analysis & Big Data

- ROOT has its own parallel processing version: PROOF the Parallel ROOT Framework
- Physics users can currently use grid and/or local PROOF clusters to analyse large datasets

- Database technology can potentially be used to do physics analysis
  - Write analysis/filtering code in SQL
- Analysis data benefits from column-storage