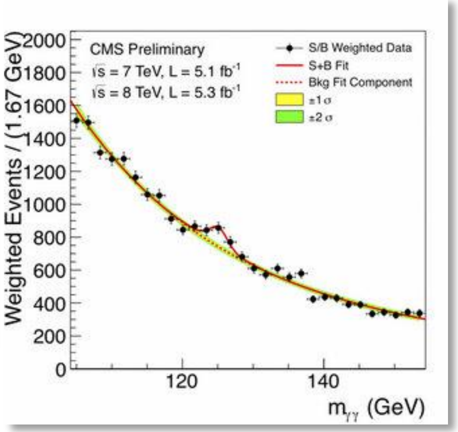
LHC Physics Analysis and Databases

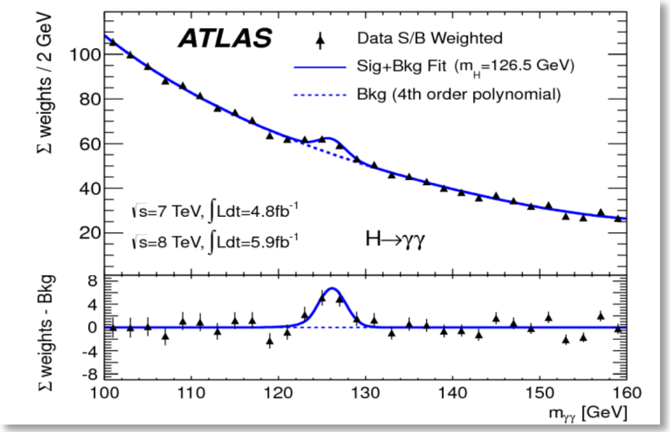
or: *“How to discover the Higgs Boson inside a database”*

Maaike Limper

Introduction to LHC physics analysis

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





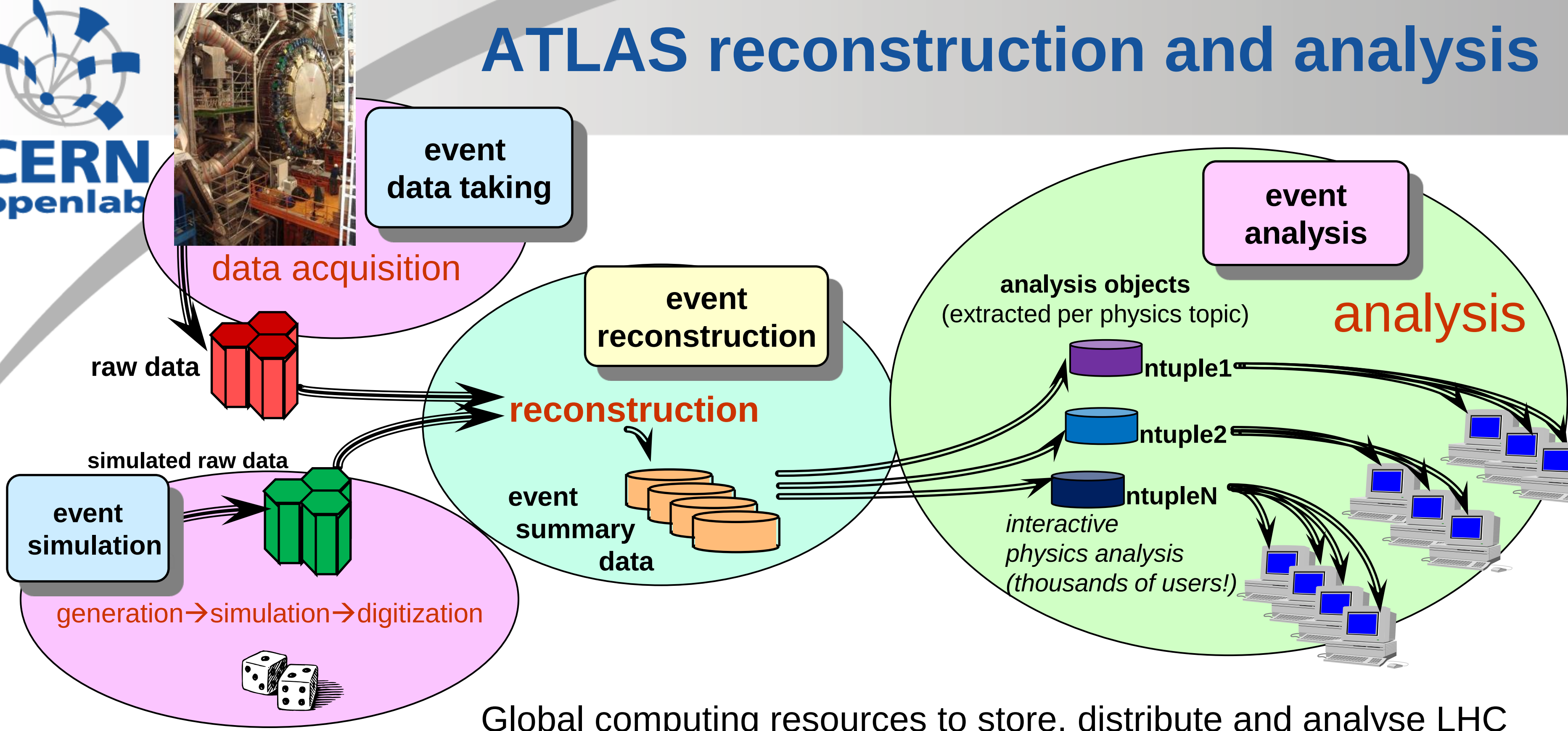
The discovery of a "Higgs boson-like" particle!

http://www.bbc.co.uk/news/world-18702455

- The work of thousands of people!
- Operations of LHC and its experiments rely on databases for storing conditions data, log files etc.

… but the data-points in these plots did not came out of a database !

ATLAS reconstruction and analysis

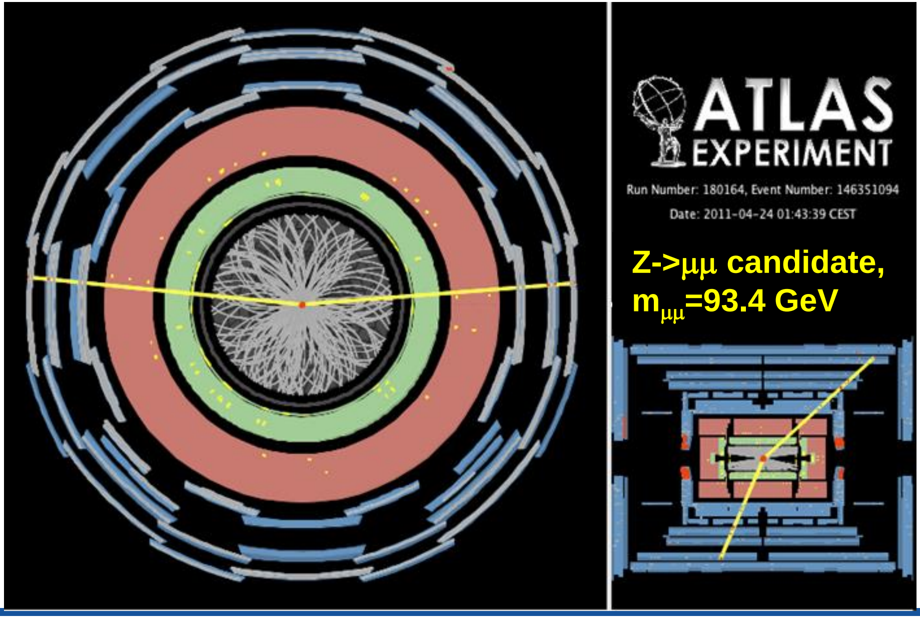


Global computing resources to store, distribute and analyse LHC data are provided by the Worldwide LHC Computing Grid (**WLCG**) which has more than 170 computing centres in 36 countries

Analysis versus reconstruction

Event Reconstruction focuses on creating physics objects from the information measured in the detector (detectors hits → particle trajectory)

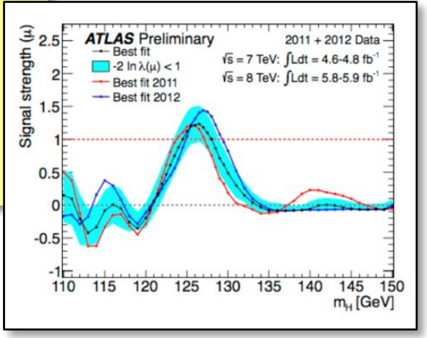
Event Analysis focuses on interpreting information from the reconstructed objects to determine what type of event took place


ATLAS EXPERIMENT
Run Number: 180164, Event Number: 146351094
Date: 2011-04-24 01:43:39 CEST
Z->μμ candidate,
$m_{\mu\mu}$=93.4 GeV

Data analysis in practice

LHC Physics Analysis is done with ROOT

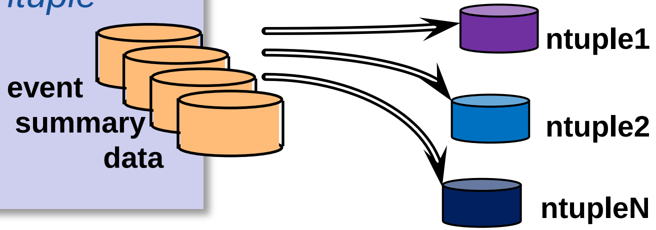
- Dedicated C++ framework developed by the High Energy Physics community, http://root.cern.ch
- Provides tools for plotting/fitting/statistic analysis etc.



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*



event summary data

ntuple1

ntuple2

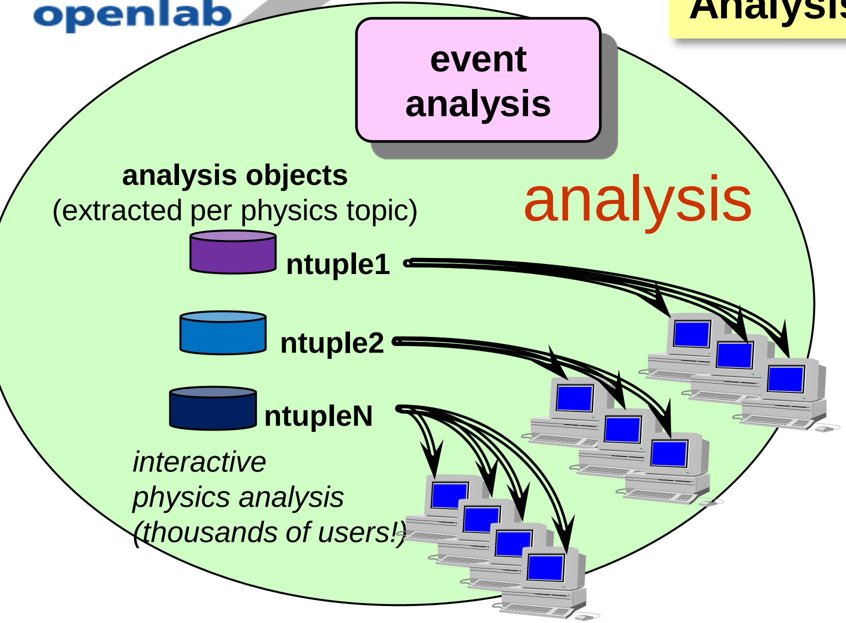
ntupleN

Ntuples=column-based storage: data is stored as “TTree” object, with a “TBranch” for each variable

Variables for each event in the form of scalar (number of muons), vectors (energy of each muon), vector-of-vectors (position of each detector hit for each muon)

Data analysis in practice

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



event analysis
analysis objects
(extracted per physics topic)
analysis
ntuple1
ntuple2
ntupleN
interactive physics analysis (thousands of users!)

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

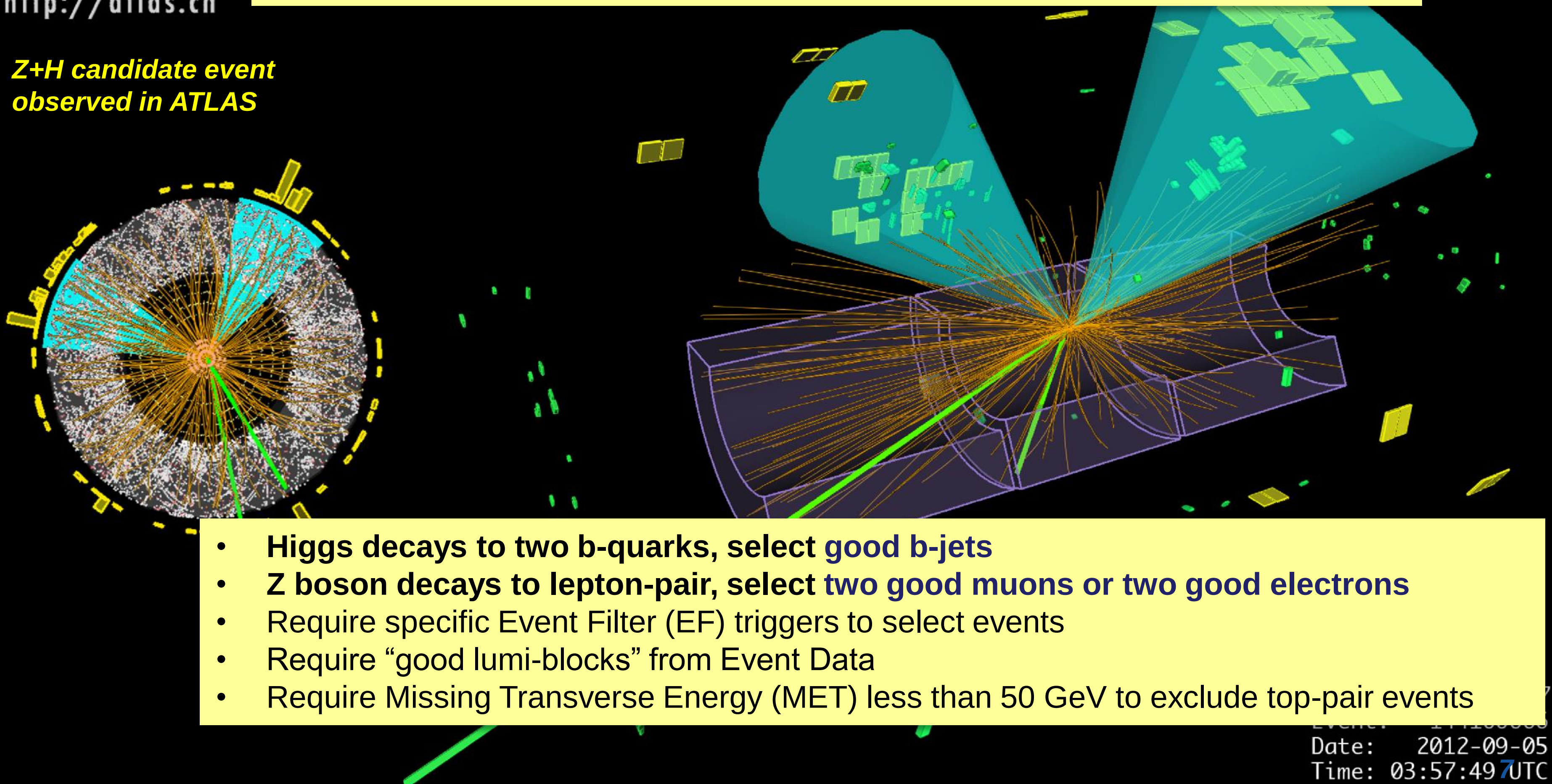
Bored waiting days for all grid-jobs to finish→ Filter data and produce private mini-ntuples

My Openlab Project: ***Can we replace the ntuple analysis with a model where data is analysed inside a centrally accessible Oracle database?***

ATLAS EXPERIMENT http://atlas.ch

My performance study for analysis inside the database used as benchmark:
"The search for a Higgs production in association with a Z boson"

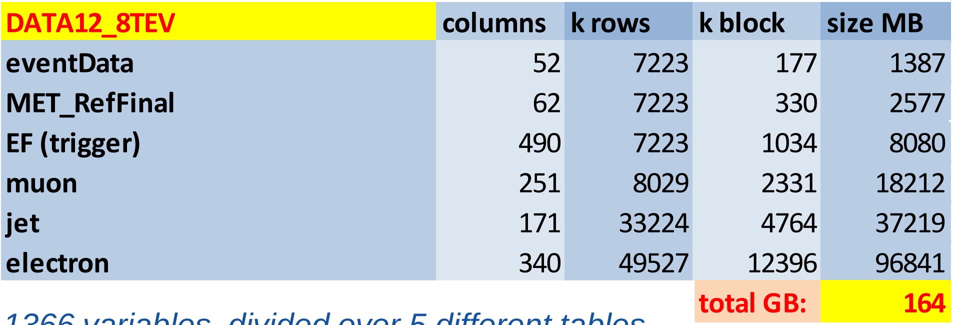
Z+H candidate event observed in ATLAS



- **Higgs decays to two b-quarks, select good b-jets**
- **Z boson decays to lepton-pair, select two good muons or two good electrons**
- Require specific Event Filter (EF) triggers to select events
- Require "good lumi-blocks" from Event Data
- Require Missing Transverse Energy (MET) less than 50 GeV to exclude top-pair events

Database design

Oracle DB has row-based storage:
Separate tables for different physics objects, so users have to read only the object-tables relevant for their analysis



DATA12_8TEV	columns	k rows	k block	size MB
eventData	52	7223	177	1387
MET_RefFinal	62	7223	330	2577
EF (trigger)	490	7223	1034	8080
muon	251	8029	2331	18212
jet	171	33224	4764	37219
electron	340	49527	12396	96841
			total GB:	164

1366 variables, divided over 5 different tables

My test-sample "DATA12_8TEV" :

- **ATLAS experiment data taken in 2012 with collission energy of 8 Tev**
- **7.2 million events**
- ~ 0.5% of all collision events recorded by ATLAS in 2012
- **Corresponds to 127 ntuple-files**

The (simplified) Z+H benchmakr analysis uses 40 of these variables
To run the analysis in the DB we need to transform a root-macro into a SQL-query

Physics Analysis C++

Root-analysis: Load relevant branches in ntuple-tree, loop over events, apply selection cuts and fill histograms:

```
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+ + ;
    }
    //etc. for muon, jet, EF selections
    //select events with 2 selected muons or 2 selected electrons and 2 good b-jets
    ….
    //after passing selection cut reconstruct invariant mass, apply combined cuts etc.
    …
    // fill histograms
}
```

Physics Analysis SQL

Single SQL-statement to reproduce physics analysis

Query starts by applying selection criteria via select-statements on relevant tables:

```
with sel_electron as
(select "electron_i","EventNo_RunNo",etc from "electron" where "pt" < 25. and abs("eta") <2.5 ···etc.  ),
sel_muon as
(select "muon_i","EventNo_RunNo",etc from "muon" where "pt" < 20. or abs("eta") < 2.4 .. etc.  ),
sel_bjet as (
select "jet_i","EventNo_RunNo”,etc from "jet" where "pt">25. and abs("eta")<2.5and mv1("wIP3D","wSV1","wFCN",..)>0.6017),
```

Followed by JOINs to find events with two b-jets and two muons or two electrons in which the invariant-mass of the electron/muon-pair and the b-jet pair is calculated and other combined selections are applied
Calculations are done via PL/SQL functions except for one function for the b-jet selection that is called from C++-library

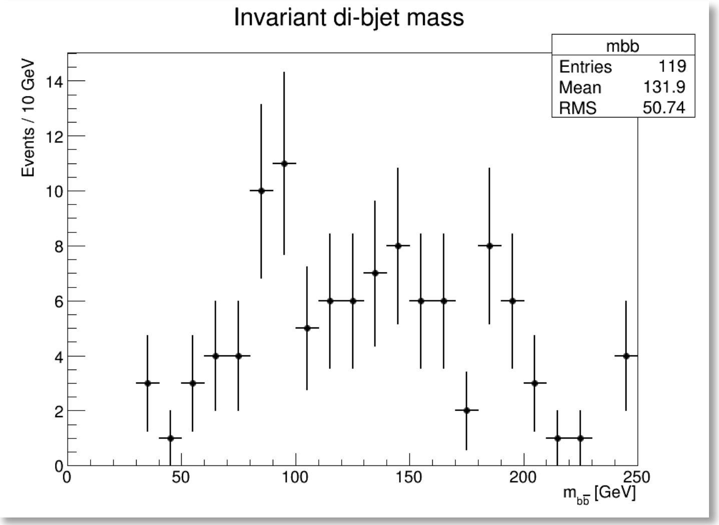
Finally a super-join where the calculated quantities, used to fill histograms, are returned:

```
select "EventNo_RunNo","EventNumber","RunNumber","DiMuonMass","DiElectronMass","DiJetMass" from
sel_muon_events FULL OUTER JOIN sel_electron_events USING ("EventNo_RunNo") INNER JOIN sel_bjet_events
USING ("EventNo_RunNo") INNER JOIN sel_MET_events USING ("EventNo_RunNo") INNER JOIN sel_EF_events
USING("EventNo_RunNo") INNER JOIN sel_goodlbn_events USING("EventNo_RunNo")
```

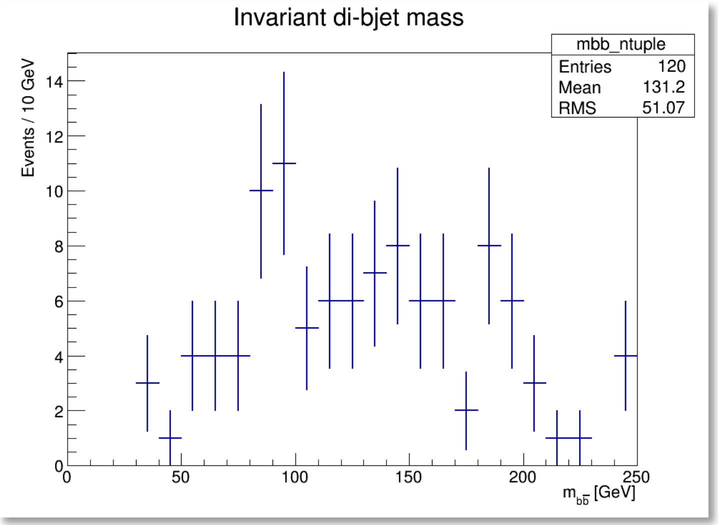
Ntuples vs DB performance

Both DB and ntuple analysis produce (almost) the same plot!

From Oracle Database



From root-ntuples



One double event in ntuple-analysis (due to overlapping trigger-streams)
Oracle DB does not contain double events due to unique constraint on EventNumber

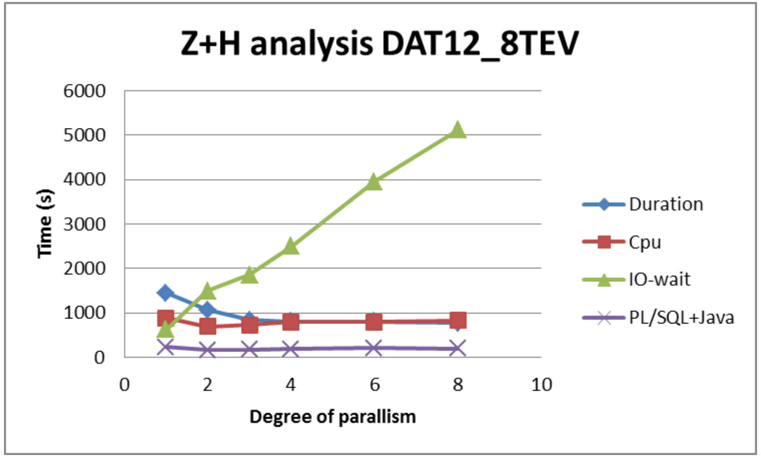
Ntuples vs DB performance

Analysis from Oracle database up to 4.5 times faster than standard ntuple analysis

- **Improvement of query time with parallel execution is limited by I/O wait time**

Benchmark analysis on "DATA12_8TEV"	**Duration (s)**	CPU time	I/O wait	PL/SQL	Java
Degree of Parallelism 1 (serial)	**1450**	890	630	21	210
Degree of Parallelism 8	**780**	840	5117	19	200
Standard root-analysis from ntuples*	3630				

*on same machine and storage as DB

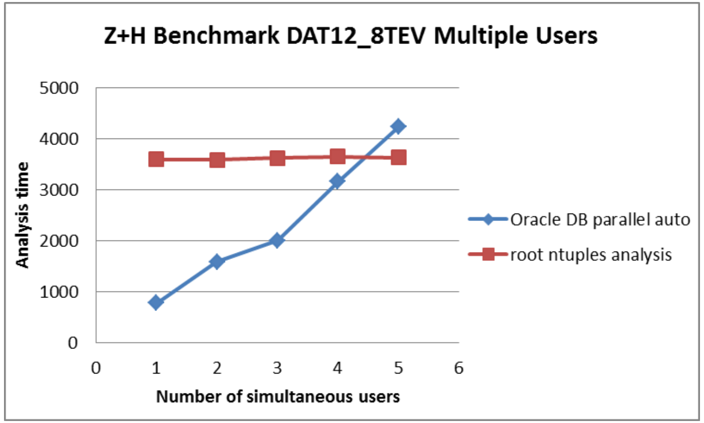


Improvement of query time with parallel execution is limited by I/O wait time!

Many Physicists!

A real physics analysis database should be able to handle multiple users accessing the same data at the same time, with each user sending a unique analysis query corresponding to the signal they are looking for

The effect of parallelism and I/O bandwidth is similar to many users accessing the same data



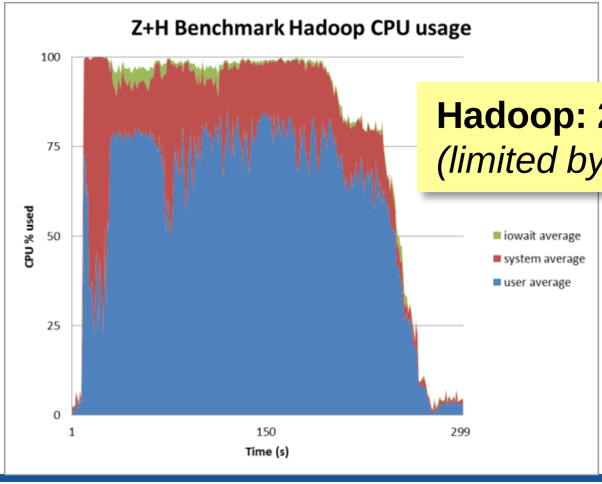
Hadoop vs Oracle

Compare the performance of the Oracle DB analysis with Hadoop!

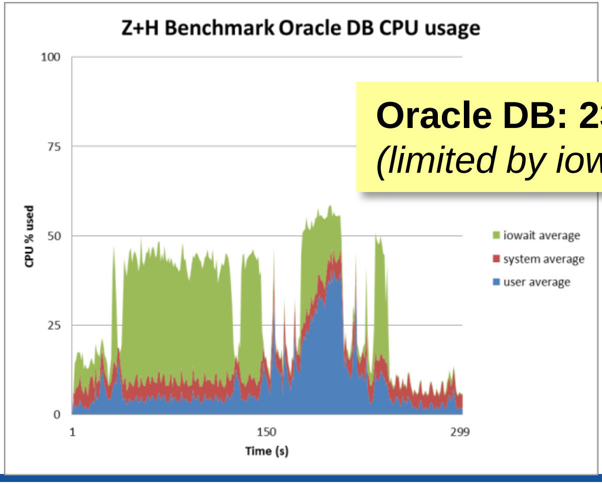
Hadoop is supposed to have fast I/O by using data-locality, I prepared a basic test:

- Physics-data stored as text-files in *hadoop filesystem (hdfs)*
- Reproduce Z+H benchmark analysis with MapReduce-code (java!)
- Mappers: one mappers per object to select muon, electron etc.
- Reduce: select events with 2 good leptons and 2 b-jets, calculate invariant mass

Tests used a 5-machine cluster that could run either Hadoop or Oracle RAC



Hadoop: 299 seconds
(limited by CPU)



Oracle DB: 231 seconds
(limited by iowait)

Conclusion/Outlook

LHC data analysis provides a real “big data” challenge

An unique benchmark to study the ability of the Oracle database to perform complex tasks on a large set of data

A central database running on a cluster of machines could provide a platform for physicists to perform analysis on data stored in the database

- **I/O is a bottleneck, especially with many users accessing the same data**
- Column-based data storage to be explored (some hints by Oracle this will be introduced in future 12c versions)
- *To be continued…*