An SQL-based approach to Physics Analysis

SELECT HIGGS FROM DATA_LHC;

M. Limper

CERN openlab

Introduction: As part of the CERN openlab collaboration an investigation has been made into the use of an SQL-based approach for physics analysis. Currently, physics analysis is done using data stored in centrally produced root-ntuples that are accessible through the LHC computing grid. We'll present an alternative approach to physics analysis where analysis data is stored in a database. This would remove the need for customized ntuple production, and allows some of the calculations that are part of the analysis to be done on the database side.

Dataset and database design:

The benchmark analysis was tested using a subset of ATLAS experiment data from root-ntuples that were centrally produced for the ATLAS top-physics group. Root-ntuples store data column-wise, while Oracle groups all related attributes together by row. A database design was chosen where physics objects were stored in separate tables.

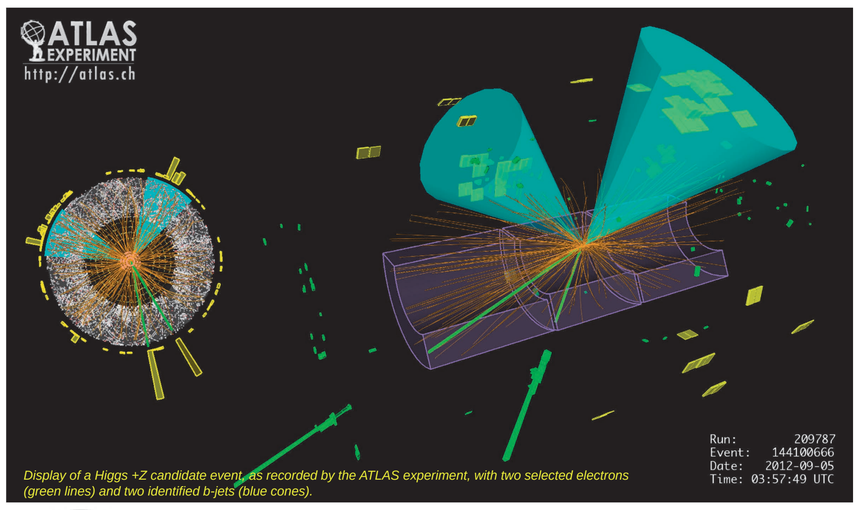
Data consistency is guaranteed through the PrimaryKey constraint on the *RunNumber, EventNumber* attributes in the *eventData*-table, which is referred to by all other tables through a ForeignKey constraint.

The table on the right shows the volume of the test-data in our database, extracted from a subset of 127 ntuples containing a total of 7.2 million events, with 4000 analysis variables per event.

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**

Benchmarks:

A simplified version of the search for the Higgs in association with a Z boson was implemented, both as a single root-macro and as an SQL-query. This analysis returns the invariant mass of the lepton- and jet-pair and uses 40 variables.



Display of a Higgs +Z candidate event, as recorded by the ATLAS experiment, with two selected electrons (green lines) and two identified b-jets (blue cones).

In addition, a cutflow analysis for the top-pair production cross-section measurement was implemented as a benchmark. In this case the original "RootCore"-packages used by ATLAS are compared to a modified set of packages that retrieve data from the DB via an SQL-query. This more realistic analysis involves 319 variables, and used data from the same tables as the Higgs+Z benchmark as well as data from the photon-table.

Physics Analysis in SQL:

The SQL-version of the benchmark analysis is built through a series of select statements on each object-table, each with a WHERE-clause to apply selection criteria. Object-selection can be done via temporary tables using the WITH-AS statement:

```
WITH goodmuons AS (SELECT … FROM muon WHERE pt>25.)
```

or by explicitly creating a table holding the objects. Materialized views can be used to define common selection criteria. For example, the benchmarks used a materialized view to define the good luminosity-block selection.

At the end of the query, JOIN statements on the RunNumber,EventNumber attributes are used to put information from the different selections together:

```
SELECT … FROM good_muons INNER JOIN good_bjets USING
(RunNumber,EventNumber) WHERE goodmuons.N=2 AND
goodbjets.N=2
```

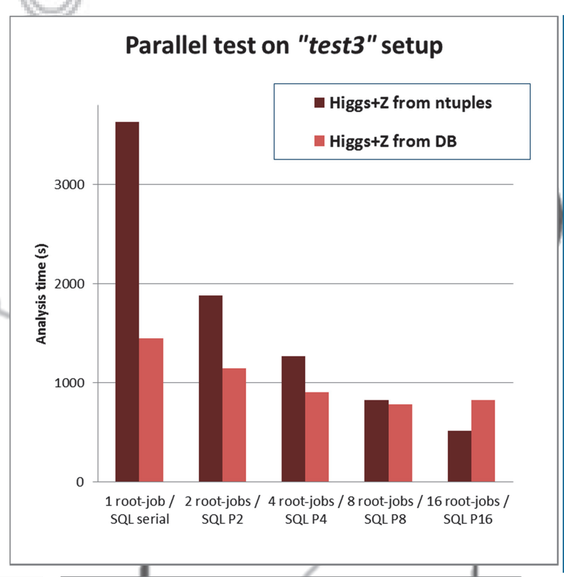
The user might be not be able (or willing) to re-write all analysis code as SQL. Simple calculations can be written in PL/SQL but one can also call exisisting C++ libraries from inside the SQL-query. In this case the external libraries need to be uploaded to the DB machines and linked to PL/SQL functions. For example, one of the external C++ libraries used by the benchmarks was used in the b-jet identification to re-calculate the b-tagging likelihood.

Test setups:

Two types of test setups were used. The "test3"-setup used 2 machines with network-based file storage (NFS) accessible from all nodes. The 2nd test setup, "mapred", was designed to run either Hadoop or Oracle RAC and was optimized for fast I/O using 5 machines connected to 5 disk arrays holding a total of 60 disks. On this test-setup the Oracle database used the Automatic Storage Management feature, and Hadoop used its hdfs filesystem, to spread the data evenly over all devices.

For the comparison with root on the *mapred*-cluster, the ntuples were distributed evenly over all disks.

Test setup	*"test3"*	*"mapred"*
# nodes	2	5
Max. I/O speed	250 MB/s	2500 MB/s
total CPU cores	32	40

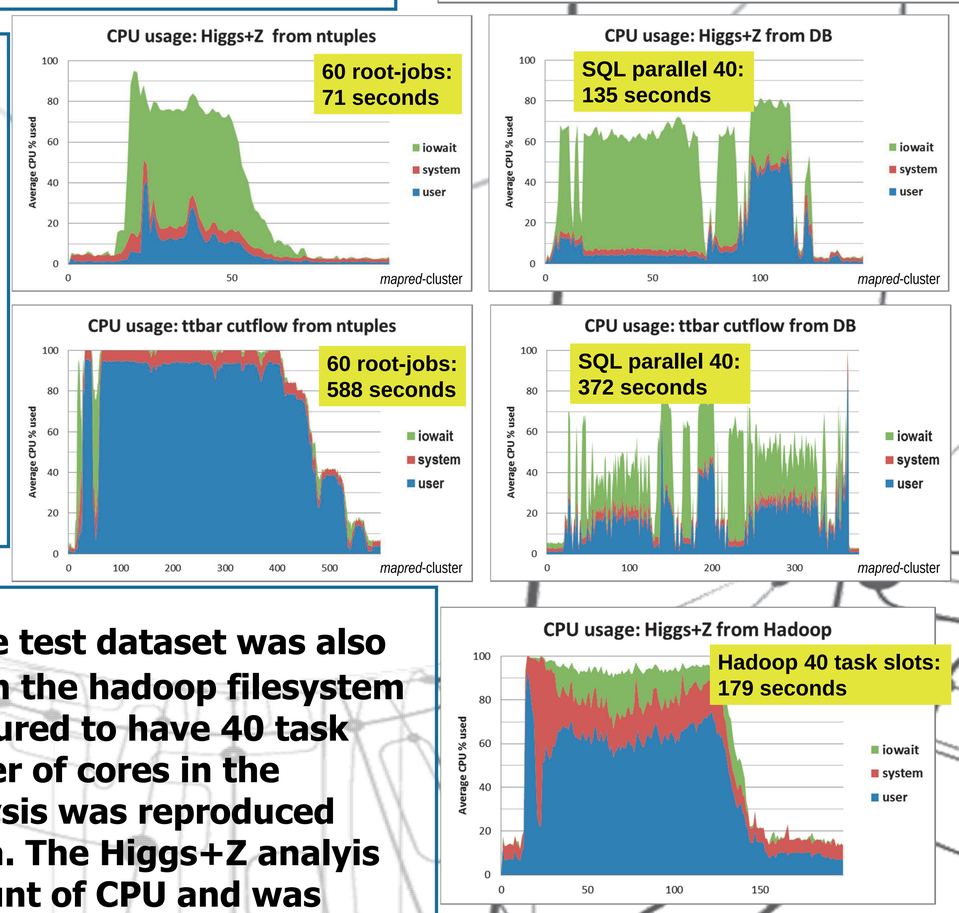


Parallel execution:

An SQL query can be executed in serial or in parallel and the degree of parallelism can be set on the table or by a hint inside the query. For the ntuple analysis, parallelism was mimicked by running multiple simultaneous root-jobs, each analysing a subset of files. The root-version gained more from parallelism than the DB-version of the analysis. This is because the DB-version is limited by I/O speed as it needs to read many columns in the table to find the relevant variables.

CPU usage:

The *mapred*-cluster was used to study CPU usage. Here, the ntuple-version was executed using 60 root-jobs (1 per disk). The plots on the right show that the Higgs+Z benchmark was fastest with the ntuple-version and both version were limited by iowait. The ttbar cutflow benchmark was faster with the DB-version as the RootCore-packages were limited by CPU.



Hadoop:

On the mapred-cluster the test dataset was also stored as comma-delimited text-files in the hadoop filesystem *(hdfs)*. The Hadoop system was configured to have 40 task slots (8 per node) to match the number of cores in the system. The Higgs+Z benchmark analysis was reproduced using MapReduce-code written in java. The Higgs+Z analyis in Hadoop used a relatively large amount of CPU and was slower than both the ntuple and DB-version.

Conclusion:

Physics Analysis using SQL on data stored in a database can provide an alternative way to analyse the large datasets produced by the LHC experiments. Row-based storage in combination with wide tables limits performance by the I/O read speed of the system. Future studies will focus on columnar stores to improve performance.