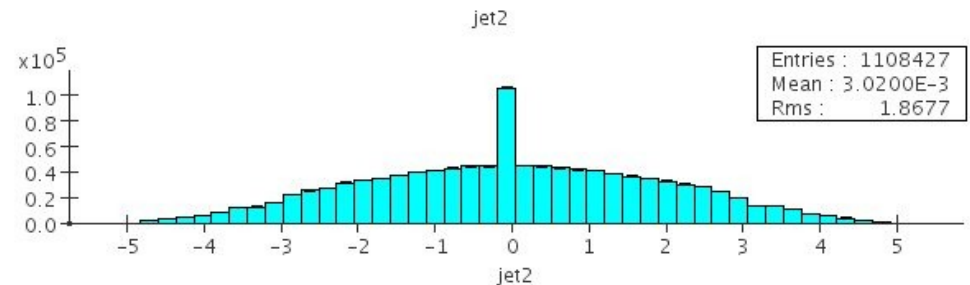
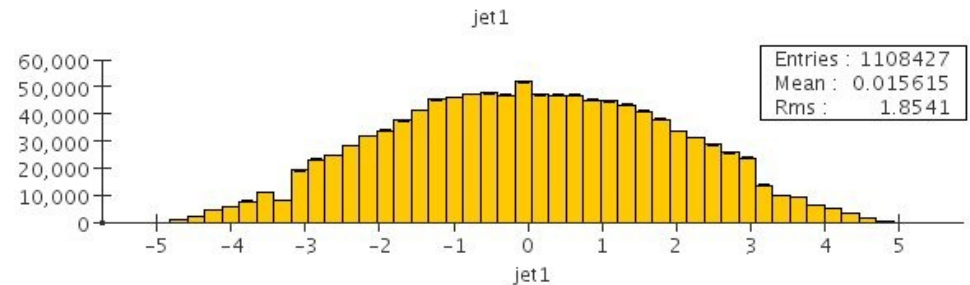


# Histo Server

- *Many technologies tried (PHP, JS,...)*
  - *It is easy just to draw a graph when you have a set of numbers*
  - *But two problems:*
    - *Our view of a histogram is very special => we have to implement ourselves all data management (accumulation, statistics, binning, limits,...) and create a set of (x,y) to draw*
    - *The drawing of axis, ticks and legends is not trivial and not handled by available packages*
- *Decision to use the standard histogramming web service package – **AIDATLD**, JSP library on top of AIDA together with standard JDBC access to Oracle*
- *As a byproduct, JSP context.xml file is created from tnsnames.ora to allow transparent Web Service access to Oracle data*

# Histo Server

- *ELSSI sends request to SQLTuple*
  - *SQLTuple calls Oracle to get data and shows them as histograms*
- *A user can access the service directly: <http://cern.ch/SQLTuple/Histogram.jsp> with appropriate parameters*
- *Testing page accessing all known data sources with various options: <http://cern.ch/SQLTuple/HistogramTest.html>*
  - *Google etc. does testing for us (a human monitor is notified by an email if anything fails)*
- *Service is distributed as SQLTuple.war file, which can be easily deployed to any Tomcat/JWSDP/... container*
- *A user can choose log/lin y axis and limit accumulated data*
  - *Other functionality can be added (choice of histo style, colors, 2d-histos, writing created histos to AFS,...)*



Download: (vector) [eps](#) [svg](#) [pdf](#) [swf](#) [ps](#) , (bitmap) [jpg](#) [png](#) [ppm](#) [gif](#)

13.2s spent - All available events analysed

Redo analysing max  events (0 means no limit)

scale:  lin  log