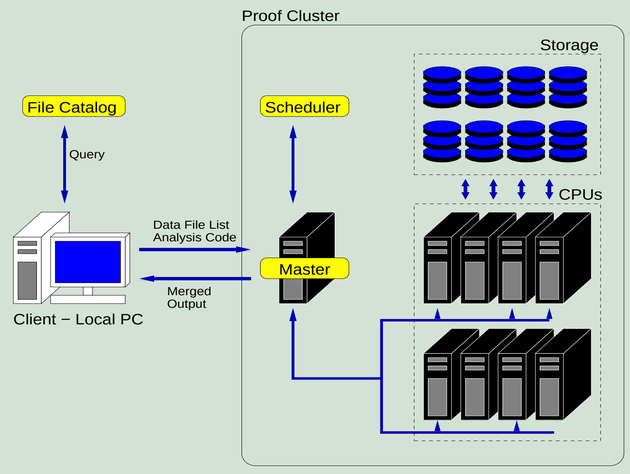
Philippe Calfayan, Ludwig-Maximilians-Universität München
Matthias Schott, CERN Geneva

# Parallel computing of ATLAS data with PROOF at the Leibniz-Rechenzentrum Munich (LRZ)

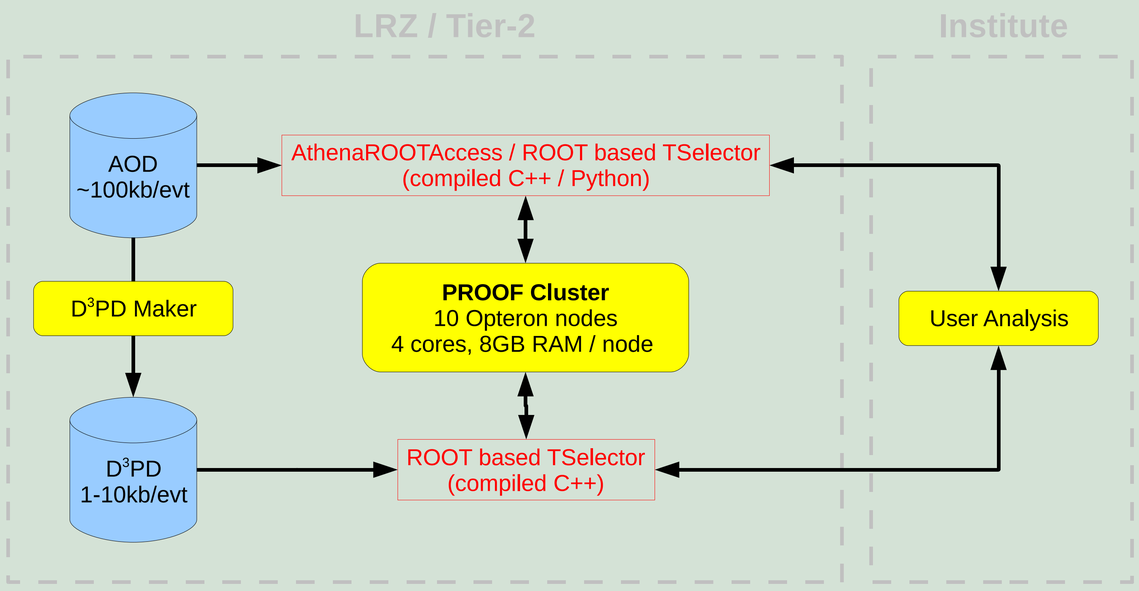
## 1. PROOF based Analyses at LRZ

**The Parallel ROOT Facility (PROOF)**

- Analyses based on TSelector (compiled or interpreted)
- Parallelisation at event level
- Use of heterogeneous nodes possible if TSelector compiled in the remote environment
- Scalability with respect to number of users and number of nodes

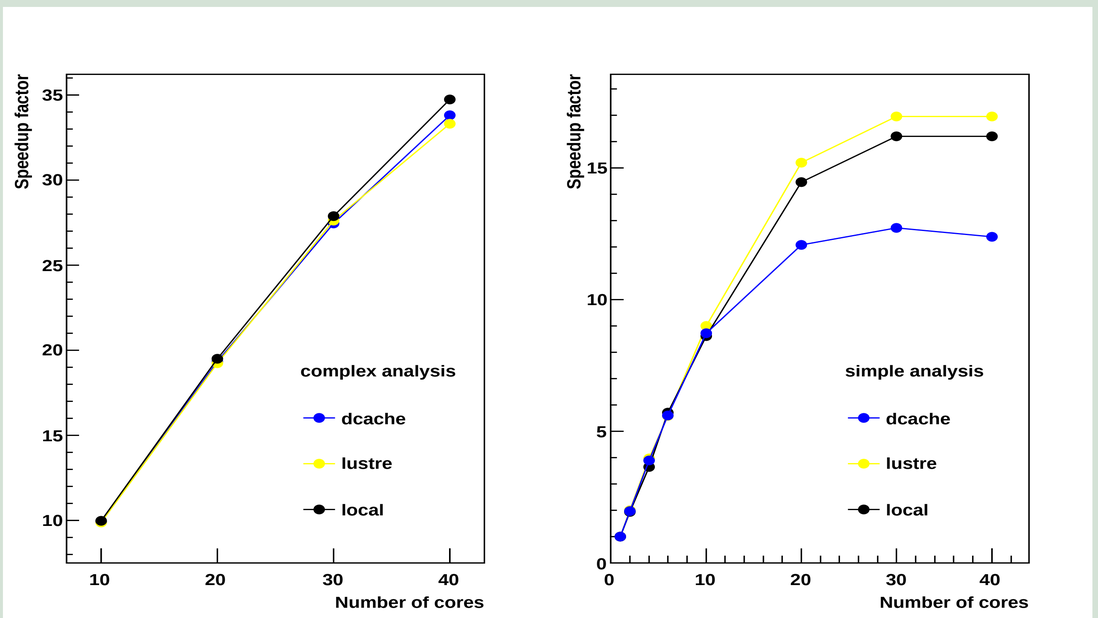



**Strategies for PROOF Analyses at LRZ**



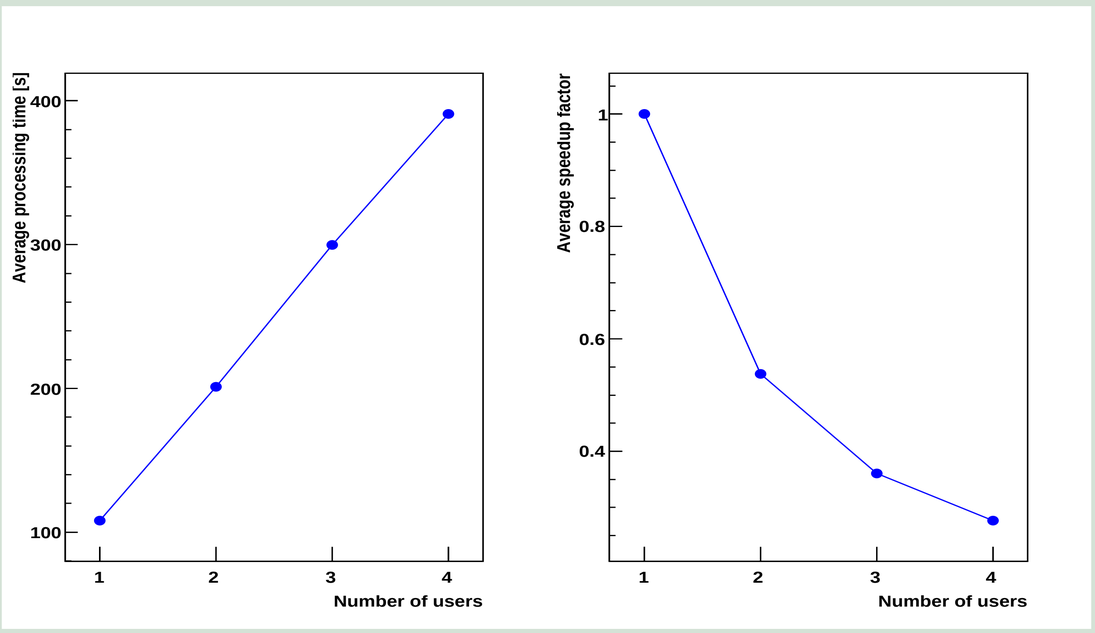

## 2. Comparison of storage strategies

- Three storage systems have been considered for the input data files:
  - → **local** disks: data are stored on each local node
  - → **dcache**: data access via client/server connections, RAID6, 10GB switch
  - → **lustre**: filesystem optimized for parallel computing, all nodes can access the data without a dedicated server
- A simple test analysis, based on the $Z$ boson reconstruction and the generation of control histograms, is processed via a ROOT based TSelector, using ROOT v5.20. A complex variant includes 200000 $\tanh$ operations per event.
- Input data files are in $D^3PD$ format (native ROOT format), and contain 1.6 million of events with a size of nearly 4kB per event.
- The speedup factor $S$ describes the gain of processing time $T_n$ using $n$ parallel cores compared to the time $T_s$ with one single core, such that $S = \frac{T_s}{T_n}$.
- The scalability of the PROOF cluster is limited by the data transfer rate of the storage systems, as shown in the figures below.




## 3. Multi-user applications

- A realistic use of PROOF would imply the management of multiple users simultaneously. Tests are carried out with the same setup as in (2).
- Only one PROOF cluster has been set up. Each user considered opens a new session using the same cluster.
- The analysis used for the tests is the complex variant of the one in (2), so that effects of the data transfer rate can be neglected.
- The Lustre filesystem has been chosen for these tests, and it is assumed that all users perform their analyses on all available cores ($n = 40$).
- Effects of potential file caching have not been prevented.
- Having $U$ users, the speedup $S$ is expected to be divided by $U$ and the time $T$ to longer by a factor $U$. The figures below confirm the scalability.
- In the plots, when $U > 1$, the time $T$ and the factor $S$ that are shown are the average of those relative to each PROOF session.




## 4. Performance with ATLAS pool files

- The ATLAS package AthenaROOTAccess allows to read ATLAS pool files (as AOD) by converting the included persistant tree into a ROOT transient tree.
- Processing AOD input files with PROOF and a compiled C++ analysis is not possible with CINT dictionaries, because of CINT limitations to handle the C++ code used in the ATLAS pool classes.
- We compiled a test analysis within the ATLAS CMT environment and generated the according REFLEX dictionary, using the Athena release v14.2.23. Two versions of the analysis are considered: one is based on a compiled C++ event loop, while the other one accesses the transcient tree with Python (via TPython).
- The test analysis runs over nearly 12500 events of a $W \rightarrow \mu\nu$ simulation (Athena v14.2.20, $\sqrt{s} = 10$ TeV), using Lustre. It calculates the $W$ transverse mass 10k times, and plots control histograms. Results are shown below.
- Comparable performances are obtained for both versions of the analysis in the case where the calculation of the transverse mass is not repeated.