

# EXPERIENCES WITH PROOF IN ATLAS

Philippe Calfayan

Ludwig-Maximilians University, Munich

DESY  
Computing Seminar  
May 4, 2009



## Outline

- The Parallel ROOT Facility (PROOF)
- Strategies for the utilization of PROOF
- Tests at the Leibnitz Rechenzentrum Munich (LRZ)
- Other tests of PROOF



## The Parallel ROOT Facility (PROOF)

### *Introduction*

- PROOF is a ROOT extension for parallel data processing, distributed with ROOT.
- Features:
  - parallelization at event level
  - integrated in the ROOT analysis framework (PROOF-based analyses depend on the “TSelector” class): moving from a local ROOT analysis to the PROOF distributed extension is transparent
  - designed for scalability: PROOF can take advantage of multi-cores laptop/desktop as well as multi-nodes clusters
  - can run interactively or in asynchronous mode
  - use of “xrootd” for communications, supports geographically distributed clusters
  - input data can be accessed through ROOT supported protocols (local or remote)
  - PROOF-based analyses can be interpreted or compiled (locally or remotely), using a cluster of heterogeneous nodes is thus possible.



## The Parallel ROOT Facility (PROOF)

### *Why PROOF?*

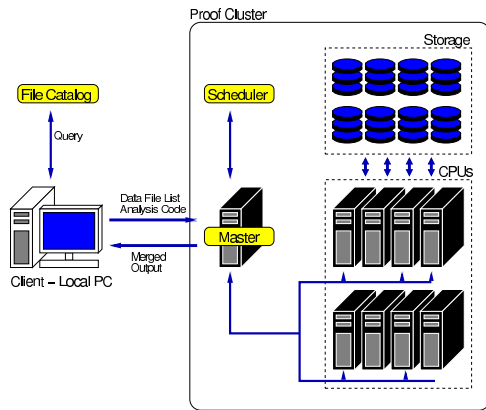
- Large amounts of data are expected to be produced ( $\sim 2.10^9$  events/year) and all events are independant from one another  
⇒ event parallelization is well suited for improving the performance of physics analyses.
- Run ATLAS analyses (using *ESD*, *AOD*,  $D^1PD$ ,  $D^2PD$ ) locally at the Institutes, with a stable and simple framework, independantly of the grid infrastructure
- Run efficiently end-user analyses with common or custom lightweight file formats ( $D^3PD$ ), independently of the ATLAS analysis framework.



## The Parallel ROOT Facility (PROOF)

### *Single master PROOF cluster*

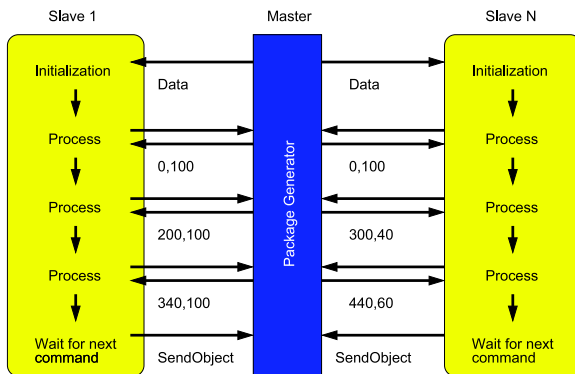
Transparency: the use of the cluster requires only minor additional configuration in comparison with a local session.



## The Parallel ROOT Facility (PROOF)

*Event parallelization: the pull architecture*

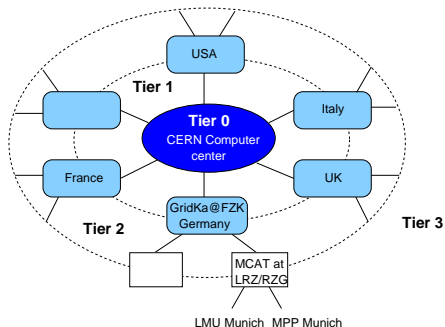
The load balancing is managed such that the workers asks for new packets to be processed whenever they are ready.



## Deployment strategies

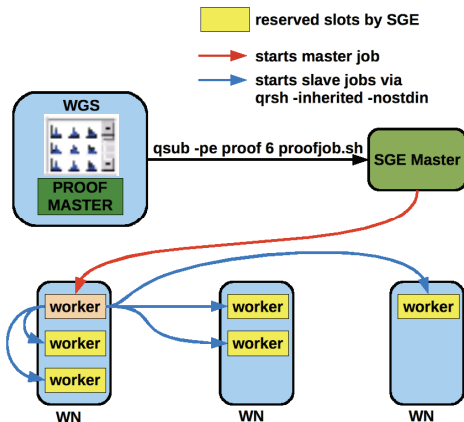
### Overview

- At Tier-1/2: configuration of a dedicated PROOF cluster together with the batch system (tests at LRZ, ATLAS)
- At Tier-1/2: configuration of a PROOF cluster on the batch system, using interactive jobs (CMS, Uni HH)
- At Tier-3/Institute: PROOF cluster on desktops/laptops (optimized with PROOF-Lite)



## Deployment strategies

*PROOF and SGE (CMS), H. STADIE and W. BEHRENOFF, CHEP09*



- Allows massive parallelization of analysis jobs
- Keep interactive "ROOT prompt"
- Used mainly by CMS (Uni HH)
- Allows for multi-user and multi-group operations
- Accounting & security possible

PROOF&SGE:  
Poster ID 66



Yves Kemp | NAF @ DESY | CHEP 24.3.2009 | Page 9



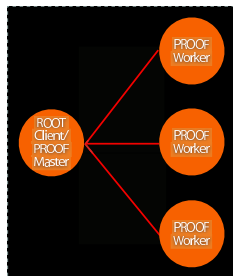
(slide from Y. Kemp, CHEP09)



## Deployment strategies

*Optimisation for multicore desktop/laptop: PROOF-Lite*

- PROOF-Lite provides a simplified interface with no cluster configuration.
- Same user code as with PROOF.



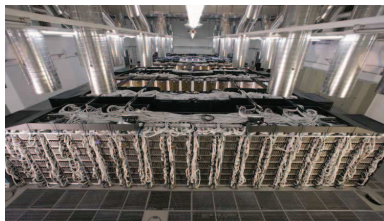
(F. Rademakers, G. Ganis, CHEP09)



## Tests at LRZ

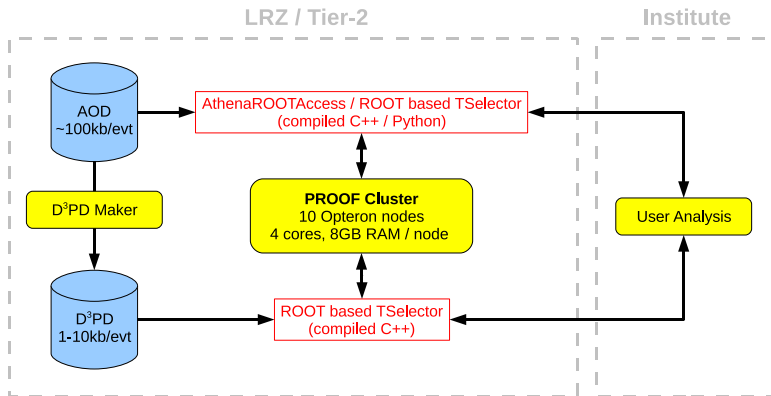
### *Setup and goals*

- LRZ: Tier-2 center
  - 10 nodes: Opteron, with dual-CPU/dual-core x86\_64 architecture (4-cores per CPU), cadenced at 2.7 GHz, and assisted by 8 GB of RAM
  - Data analysis, scalability tests, I/O and CPU performance
- Results presented at CHEP09 (P. Calfayan)



## Tests at LRZ

*Analysis strategies: two methods to exercise PROOF*



## Tests at LRZ

### *Measure of performance*

- The total execution time with  $n$  workers can be expressed as follows:

$$T_{total,n} = T_{init} + T_{data\ transfer} + \frac{T_{events\ process}}{n} + T_{post\ process}$$

- For these tests, we measure:  $T_n = T_{total,n} - T_{init}$

The initialization time of the cluster is not taken into account.

- The speedup factor  $S_n$  describes the gain of processing time  $T_n$  using  $n$  parallel cores compared to the time  $T_1$  with one single core, such that:

$$S_n = \frac{T_1}{T_n}$$

- The processing time of the last packet determines the total execution time (due to the pull architecture).



## Tests at LRZ

### *Storage strategies: introduction*

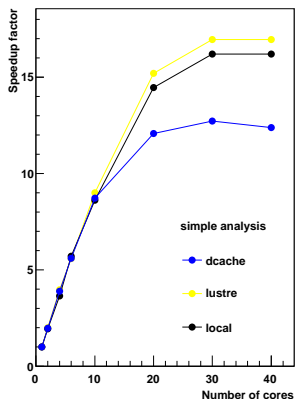
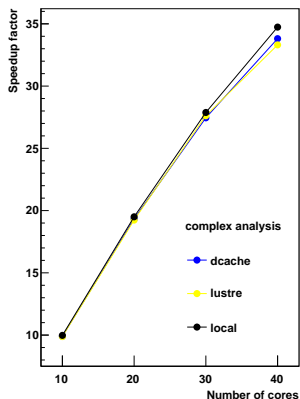
- Three storage systems have been considered for the input data files:
  - **local** disks: data are stored on each local node  
For  $n > 10$  nodes, the data transfer rate is limiting due to concurrent access.  
(reading rate of disks:  $\sim 35$  MB/s)
  - **dCache**: data access via client/server connections. Storage disks combined with RAID6 array, and accessed through 10 GB switch.
  - **lustre**: filesystem optimized for parallel computing. The lustre disks are accessible from all working nodes, 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.



## Tests at LRZ

### *Storage strategies: results*

The scalability of the PROOF cluster is limited by the data transfer rate of the storage systems in the case if the simple analysis.



## Tests at LRZ

### *Multi-user tests: introduction*

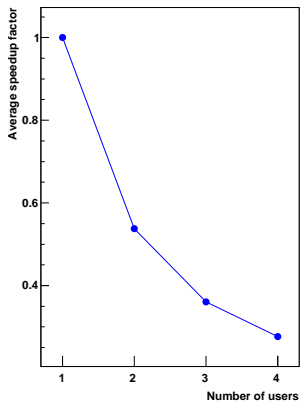
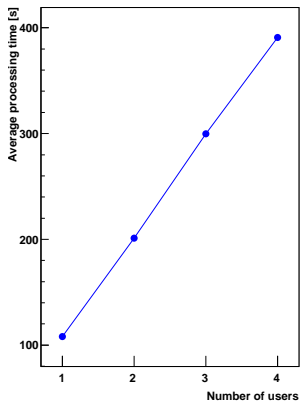
- A realistic use of PROOF would imply the management of multiple users simultaneously.
- Only one PROOF cluster has been set up. Each user considered opens a new session using the same cluster.
- The analysis used for the tests is the complex variant of the one considered for the storage tests, so that 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 be longer by a factor  $U$ .



## Tests at LRZ

### *Multi-user tests: results*

- The figures below confirm the scalability w.r.t. number of users.
- For  $U > 1$ , the time  $T$  and the factor  $S$  are the average of those relative to each PROOF session.



## Tests at LRZ

### *Running on AOD files with PROOF: introduction*

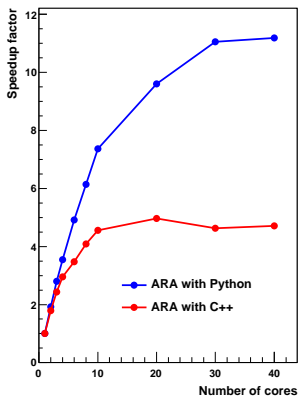
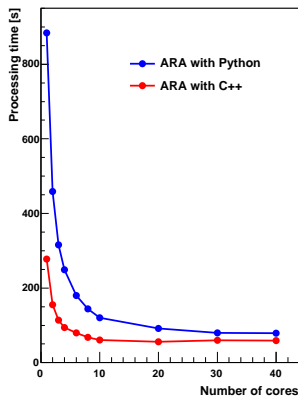
- We use AthenaROOTAccess to read AODs (persistent tree  $\rightarrow$  ROOT transient tree).
- Processing AOD pool input files with PROOF and a compiled C++ analysis is not possible with CINT dictionaries, because of CINT limitations.
- We compile the analysis loop (TSelector) in a CMT package with Athena 14.2.23, and use a REFLEX dictionary.
- Transient tree read in 2 ways: compiled C++ or Python (via TPython in a compiled TSelector).
- Nearly 12500  $W \rightarrow \mu\nu$  simulated events are processed (generated using Athena 14.2.20 and  $\sqrt{s} = 10$  TeV)
- The test analysis calculates the  $W$  transverse mass 10k times and plots control histograms.
- Input files are stored using Lustre.



## Tests at LRZ

*Running on AOD files with PROOF: results*

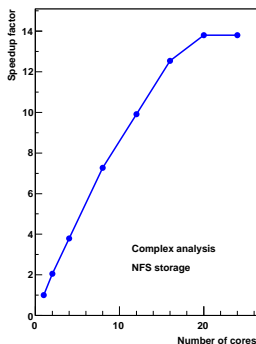
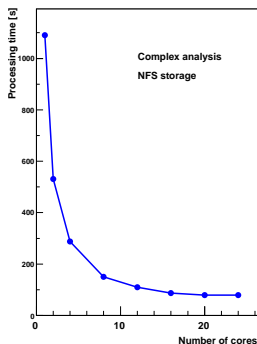
The compiled C++ version reaches the I/O bound before its Python counterpart.



## Tests at the Ludwigs-Maximilians University Munich

### *Example of end-user analysis at Institute: scalability tests*

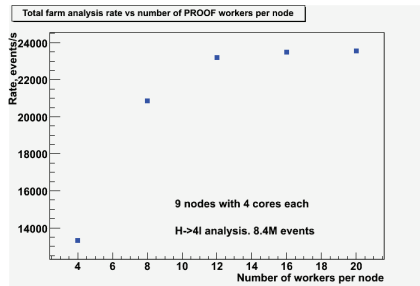
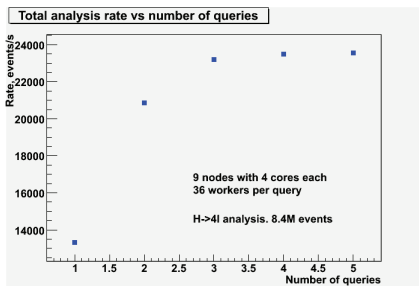
- The PROOF cluster set up at the occasion of the test is composed of 6 Core2 Quad CPU running at 2.8 GHz, and assisted by 8 GB of RAM (up to 24 cores).
- Input data accessed by NFS mounted disks
- Complex test analysis (Leptoquark search) using a multivariate technique.
- Input data in custom ROOT format ( $\sim 1$  KB/event ).



## Tests at the Brookhaven National Laboratory (BNL)

*Scalability tests, Sergey Panitkin (CHEP09)*

- PROOF test farm co-located with Tier-1
- Connected with dCache via xrootd

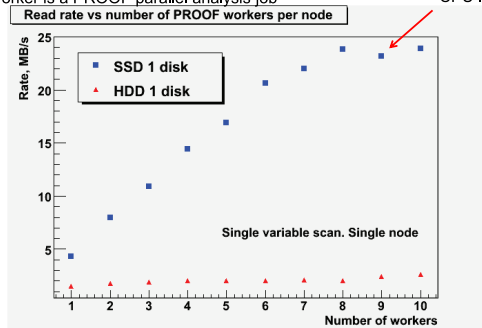


## Improving the I/O limitation

*Test of SSD disks with PROOF, BNL, Sergey Panitkin (CHEP09)*

- Model tested: 64 GB, acces time  $\sim 0.1$  ms (vs  $\sim 10$  ms for typical SATA disks), reading speed:  $\sim 120$  MB/s

- Worker is a PROOF parallel analysis job



- SSD holds clear speed advantage
- $\sim$ Up to 10 times faster in concurrent read scenario



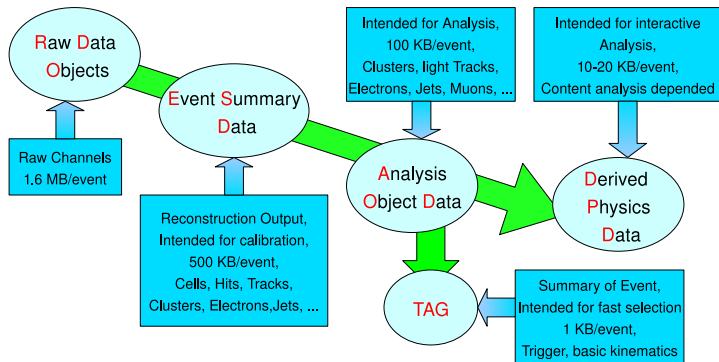
## Conclusion

- PROOF is well suited for complex physics analyses at hadron colliders.
- Main issue is related to the I/O bound of the data storage.  
SSD disks could be a solution.
- Several deployment strategies are possible: dedicated server or together with interactive batch jobs.
- Can be used directly over ATLAS pool files (AOD) (via AthenaROOTAccess + REFLEX dictionary) or for end-user analysis on files in native ROOT format ( $D^3PD$ ).
- PROOF-Lite optimizes end-user analyses running on single multicore machine.



## Backup

### *ATLAS event data model*



(sketch from J. Elmsheuser, CHEP09)

