PROOF Status

G. Ganis, CERN

ALICE Offline week, 14 July 2011

Outline

- What's new in ROOT 5.30 wrt 5.28
- Current development

News in 5.30 wrt 5.28

- New benchmark suite (TProofBench; S. Ryu)
- Integration w/ afdsmgrd (D. Berzano) **5.28d**
- Support for file access via the PROOF port **5.28c**
- Support for log file truncation **5.28c**
- Added dataset and memory info in monitoring and possibility to send to multiple collectors **5.28c**
- Support for protocol 'pod://' to start PoD

Improvements / fixes in 5.30

(mostly included in 5.28e)

- TProof::ClearPackages: enforce execution on all nodes **5.28-ptc**
- TProof::Load: support for multiple-headers **5.28a**
- TProof::GetDataSets: support for a fast lite version **5.28a**
- Export envs ROOTPROOFCLIENT, ROOTPROOFLITE **5.28a**
  (steer building/enabling of PARs in BUILD.sh/SETUP.C)
- Fixes in the monitoring interface **5.28c**
- Full support for temporary/data files on shared FS **5.28c**
- Correct treatment of TSelector::kAbortFile **5.28c**

Improvements / fixes in 5.30

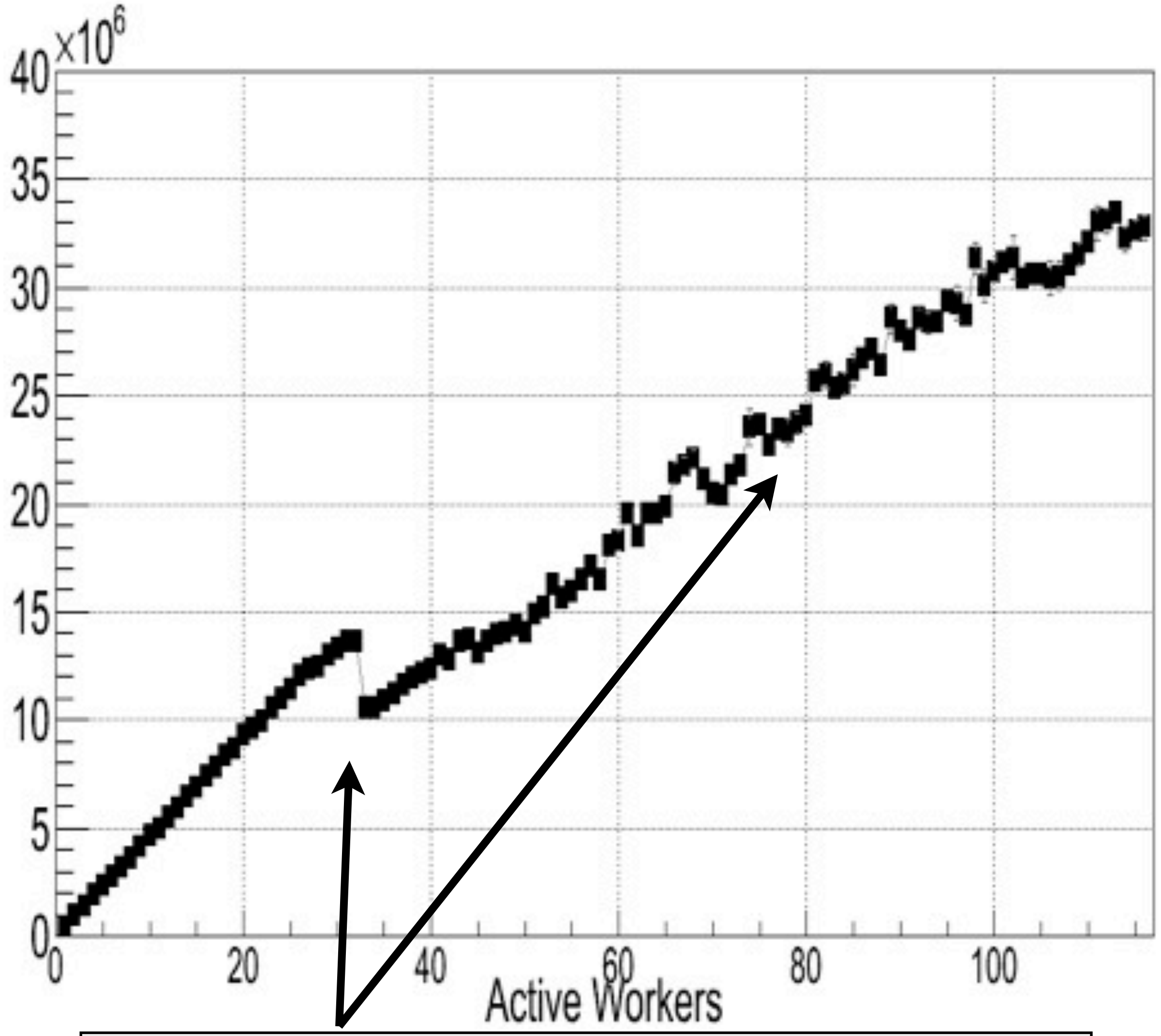
(cnt'd)

- Improved reporting of missed {files, events} 5.28c
- Improved logging of failures 5.28c
    - Log file availability and completeness
- TPacketizerUnit: improved packet algorithm 5.28d
- Use TH1::Add instead of TH1::Merge 5.28-ptc

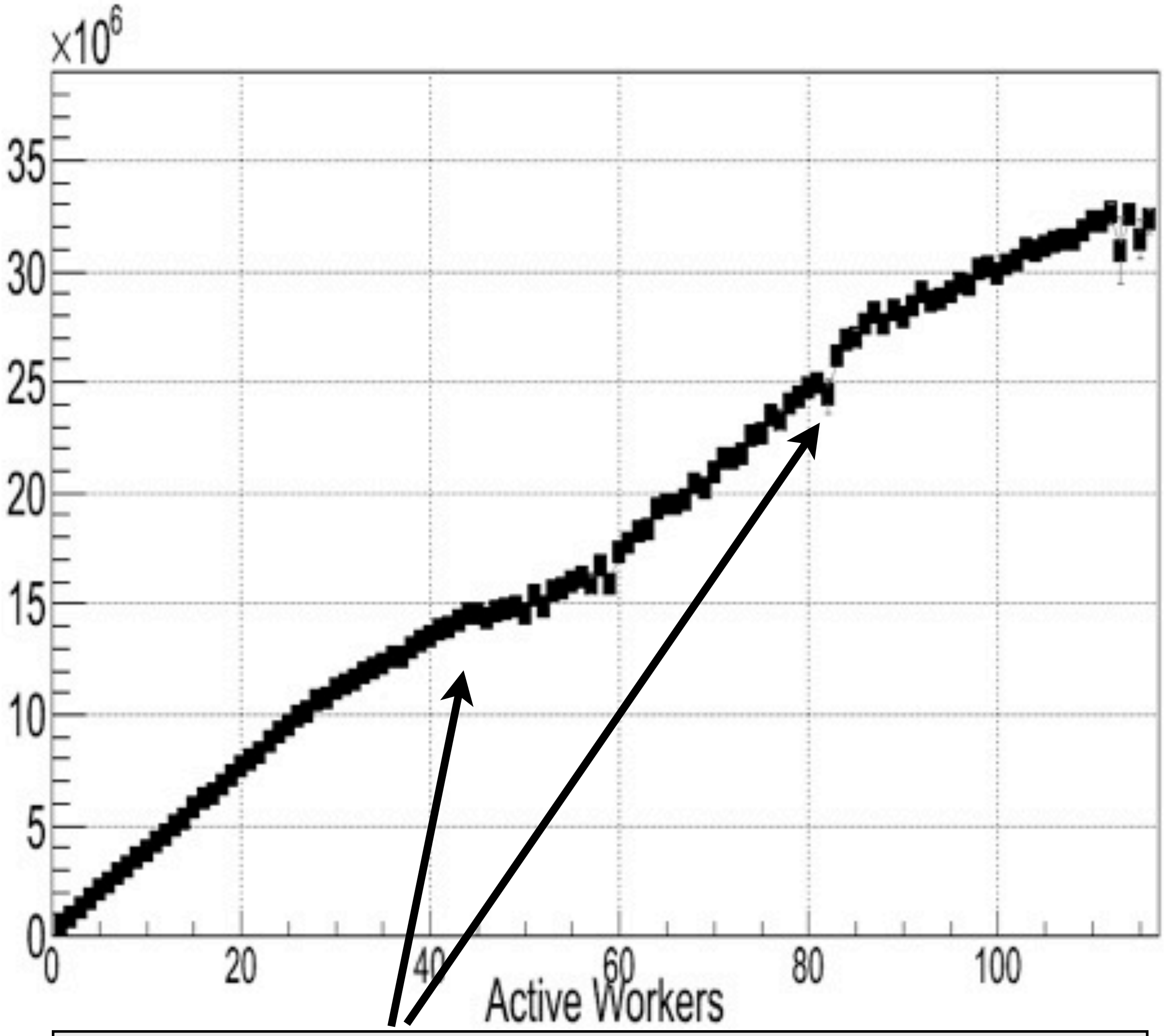
TPacketizerUnit fix

5.28d

CPU benchmark (TProofBench) on CAF (3 types of cores, different performance)



Broken packetizer does not follow well the addition of a less performant CPU



Fixed packetizer moves smoothly between performance regimes

5.28-ptc

5.30(a)

Merging using TH1::Add

Simple
test macro

```
void testUseH1Merge(const char *master = "localhost", Bool_t useMerge = kFALSE) {
   if (!master) master = "localhost";
   TProof *p = TProof::Open(master);
   if (useMerge) p->SetParameter("PROOF_UseTH1Merge", 1);
   p->SetParameter("ProofSimple_NHist3", (Long_t)50);
   TStopwatch tt;
   p->Process("tutorials/proof/ProofSimple.C+", 1000000);
   tt.Print();
   TStatus *st = (TStatus *) p->GetOutputList()->FindObject("PROOF_Status");
   if (st) st->Print();
}
```

```
root [0] .x testUseH1Merge.C("localhost", kTRUE)
...
PROOF set to parallel mode (8 workers)
Mst-0: merging output objects ... done
Mst-0: grand total: sent 157 objects, size: 212365610 bytes
Real time 0:00:42, CP time 3.500
OBJ: TStatus    PROOF_Status    OK
 Max worker virtual memory: 394.91 MB   Max worker resident memory: 229.43 MB
 Max master virtual memory: 2295.74 MB  Max master resident memory: 2139.77 MB
```

Using TH1::Merge
(scales w/ N_{wrk})

Using TH1::Add :
40 % faster,
< 50% RAM
(~ constant w/ N_{wrk})

```
root [0]  .x testUseH1Merge.C("localhost")
...
PROOF set to parallel mode (8 workers)
Mst-0: merging output objects ... done
Mst-0: grand total: sent 157 objects, size: 212365610 bytes
Real time 0:00:26, CP time 3.520
OBJ: TStatus    PROOF_Status    OK
 Max worker virtual memory: 394.90 MB   Max worker resident memory: 229.47 MB
 Max master virtual memory: 1104.55 MB  Max master resident memory: 940.73 MB
```

Current developments

- (Dynamic) consolidation
- Towards Multi-Master setups
- Packetizer restructuring

(Dynamic) consolidation

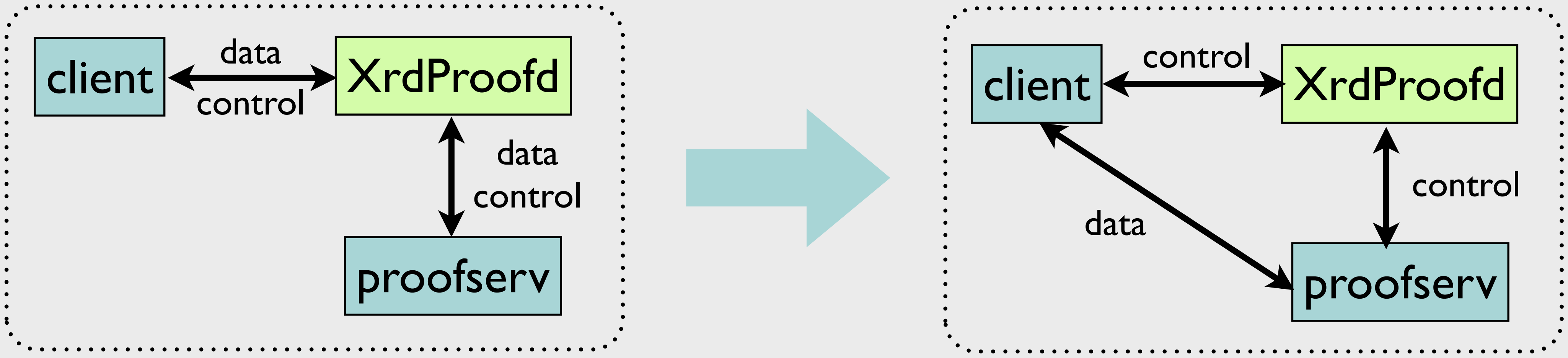
- Bug fixing
- Correct unnatural/illogical/user-unfriendly behaviors with targeted developments
  - New connection layout w/ direct connection
    - Remove interference between users
  - Automatic merge optimization
    - Better handling of merging problems

5.31/02
end August ?

Connection layout

~~**5.29/02**~~
~~**end March**~~

- Master redirection
- External startup via system() instead of fork() **5.30**
  - Speed-up session startup
- Direct data connection



  - Remove Single-Point-Of-Failure

G.Ganis, PROOF Status, Alice Offline Week, 11 March 2011

Automatic merge optimization

- Automatic switch to merge-via-file if output is large
- Present output objects to users as they were in the output list
  - TSelector::FindOutputObject
    - Take it from the fOutputList or from the file
- Optional: delayed merging or leave files un-merged

Multi-Master setups

- Dynamic re-organization of a set of workers in a multi-tier structure
  - Use PROOF-Lite for nodes with $N_{workers} > 1$
- MM dataset manager
- MM-Packetizer for dynamic load-balancing across sub-masters
- Support for non-data driven tasks

Packetizer issues

- Problem with TPacketizerUnit shows the importance of good packetizing
- TPacketizerAdaptive is in some cases *slower* than the old and trivial TPacketizer
- TPacketizerAdaptive difficult to maintain
  - Author unavailable; complicated algorithm

Packetizer redesign

- Full understanding and control of the behavior in most typical cases
  - Large number of events of ~same size (analysis)
  - Small number of events of different size (reconstruction)
  - Non-homogeneous distribution

Packetizer new features

- Ability to dynamically handle new work, i.e. new files
  - Needed by the MM-packetizer
- Thread-awareness
  - Needed to deserialize packet assignment on master

Both needed to scale to bigger clusters

New acquisition

- Anar Manafov joined the PROOF dev team
- First assignment: the new packetizer

$$$

Other issues

- Dataset verification in parallel
- Support for processing a TSelector by object
- Full draw functionality via TTreePlayer
- Further improvements in monitoring
- ...

List of tasks maintained in Savannah ...

Monitoring

- Better classification of dataset information
  - Separate table, in SQL terms
- Option to send detailed information about files (ATLAS request)
- Internally, additional abstraction level for the monitor sender to better handle different backend requirements

Documentation

- New ROOT userguide in Docbook
  - In SVN: submodule $ROOTSYS/doc
- There will be a chapter on PROOF ...

Questions ?